Likelihood-free Bayesian Inference for Dynamic, Stochastic Simulators in the Social Sciences

Joel Dyer

St Hilda's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2022

Abstract

Simulation models – such as agent-based models (ABMs) in the social sciences – are now used widely across scientific and commercial domains. However, such models often lack a tractable likelihood function, precluding standard likelihood-based statistical inference. In response to this challenge, the past two decades have seen the development of likelihood-free, simulation-based procedures for inferring simulator parameters within the computational statistics and machine learning communities. A prototypical and theoretically appealing approach is approximate Bayesian computation (ABC), in which the pertinence of parameter values is determined on the basis of a meaningful notion of distance between the observed data and output generated by the simulator at those parameter values. However, ABC typically requires many hundreds of thousands of calls to the simulator to construct accurate posterior densities, making it unsuitable for computationally expensive simulators such as macroeconomic ABMs. Furthermore, there are few ABC approaches – and likelihood-free inference (LFI) approaches more generally – that are compatible with generic time-series simulators: many approaches involve reducing the data to hand-crafted summary statistics – which can require substantial domain expertise and can lead to a deleterious loss of information – or making inappropriate assumptions, such as independent and identically distributed (*iid*) or regularly spaced observations.

In this thesis, we aim to develop the literature on likelihood-free inference algorithms for generic time-series simulators, with a focus on simulation models in economics and the social sciences. To this end, we present the following contributions:

Section 1: Introduction In the first section of this thesis, we introduce the basic problem of (Bayesian) statistical inference, and the problem of extending this to generic simulation models for which standard likelihood-based inference procedures are not immediately available. We then provide a review of the literature in this area, both as it appears in the computational statistics community and the social sciences, before setting out the aims and objectives of the thesis.

Section 2: Approximate Bayesian Inference with Path Signatures In the second section, we will introduce the use of the *path signature* as a natural, automatic

feature set for approximate Bayesian inference algorithms involving generic time-series simulators. Besides introducing path signatures, this section will consist of three main chapters. In the first of these, we will motivate and present experiments on the use of path signatures as automatic summary statistics in ABC. In this way, we will demonstrate that signatures permit a natural notion of distance between sequential data and a powerful means to performing so-called semi-automatic ABC, with competitive empirical performance in a set of benchmarking experiments. In the second and third, we will consider the problem of performing density ratio estimation (DRE) – an alternative approach to LFI – using path signatures. In particular, we will investigate their use in neural DRE methods in Chapter 4 as a natural means to learning low-dimensional, approximately sufficient summary statistics for sequential data; we will then demonstrate in Chapter 5 that their use in kernel logistic regression is able to yield more accurate parameter posteriors relative to competing methods in low–simulation-budget regimes.

Section 3: Black-box Neural Posterior Inference for Agent-based Models In this section, we will concentrate more specifically on parameter inference for agent-based models, which is a class of simulation models that is growing in popularity in economics and the social sciences. We will argue for the use of neural DRE and a further class of neural simulation-based inference methods – neural posterior estimation (NPE) – in parameter inference tasks for complex and expensive simulation models such as ABMs. To do so, we will present experiments in which we demonstrate the greater simulation efficiency and accuracy of DRE and NPE, and thus their potential as generic inference procedures for expensive ABMs in economics and the social sciences. Finally, we will discuss how such approaches naturally extend to less typical but more complex inference tasks that may be encountered in ABM settings, such as when sequences of graphs – rather than the more typically encountered sequences of Euclidean data – are observed. This can be the case when e.g. dynamic social networks or occupational mobility networks in labour markets are the target of the ABM. Our contributions demonstrate that these methods are better tailored to the key challenges faced when calibrating arbitrary ABMs to data than the current most popular methods in the social scientific ABM literature, and that these methods enable agent-based modellers to automatically perform parameter inference for ABMs that model complex and high-dimensional temporal datasets.

Acknowledgements

I have many people to thank for making the work presented in this thesis possible, and for enriching my research experience over the past 4 years. I explicitly name many but not all of these people below.

First, I would like to thank my supervisor, J. Doyne Farmer, for his support and guidance throughout my DPhil. I'm incredibly grateful for the opportunities he has given me, the freedom he provided for me to develop my own ideas and to be creative, and for coming to the rescue more times than should have been necessary.

Secondly, I would like to thank the Research Team at Improbable for having supported and sponsored my DPhil over the past 3.5 years. In particular, I have been very fortunate to work closely with, and to have received further supervision from, Sebastian M. Schmon and Patrick Cannon during this time – thank you both for helping to make my research studentship far more productive and enjoyable than it would have otherwise been.

Third, I would like to thank the various academics, students, and funding sources that enabled me to carry out my research degree. I gratefully acknowledge EPSRC and the Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) for having supported this research. Particular thanks go to Chris Breward and Colin Please for directing the CDT. I also thank the Complexity Economics group, and INET Oxford more broadly, for having been an academic home for me over the past 3 years. Further thanks go to Horatio Boedihardjo, Sam Cohen, Lajos Gergely Gyurko, Zacharia Issa, Blas Kolic, Renaud Lambiotte, Terry Lyons, James Morrill, Harald Oberhauser, Cristopher Salvi, and the students and staff of the InFoMM CDT for their comments, feedback, friendship, helpful discussions, and informal support throughout the past 3 years.

Fourth, I would like to thank The Alan Turing Institute and the Jean Golding Institute at the University of Bristol for the opportunities and connections made available to me through the Alan Turing Institute's Enrichment Scheme. Particular thanks goes to Anya Skatova for hosting me throughout the scheme, and for her generosity and the very useful discussions we had.

Finally, to those closest to me:

Mum – there is so much to thank you for and not enough space here. I'm lucky to have you – thank you for everything.

Simon – thank you for your friendship, for your support, and for taking care of us all over the years.

My sister – thank you for being a role model for me. Navigating the difficult times would have been far harder without your advice and example to help along the way.

Miskina – thank you for your warmth, care, and encouragement over the past three years. This thesis is only the second best thing to have come from my time at Oxford.

My brother – my best friend and *raison d'être*. Thank you for all the mischief and laughter.

Declaration

This thesis is submitted to the University of Oxford in support of my application for the degree of Doctor of Philosophy. It has not been submitted in any previous application for any degree. The work presented (including data generated and data analysis) is the author's own except for the cases outlined below:

Abstract The abstract was written by the author.

Part I The entirety of Chapter 1 up to and including Section 1.4 was written by the author. Sections 1.5.1 and 1.5.2 were written by the author, incorporating feedback from Patrick and Sebastian, with Patrick contributing additional detail in Section 1.5.2.3 after it was originally drafted by myself. Sebastian contributed the first draft of Section 1.5.3 which I later expanded on. Section 1.5.4 was written by the author, incorporating feedback from Patrick and Sebastian. Sections 1.5.5 and 1.5.6 were written by the author. Sections 1.6 and 1.7 were written by the author.

Part II Chapters 2–5 are derived from work the author carried out with Sebastian M. Schmon and Patrick Cannon. The following provides a more detailed breakdown of the contributions:

Chapter 2 This chapter was written by the author.

Chapter 3 The original idea for this chapter was the author's, and was subsequently developed further and jointly with Sebastian and Patrick. I implemented the methods, developed and conducted the numerical experiments (with the exception of the particle filter method for obtaining the ground-truth posterior for the Ricker model, which was

implemented by Sebastian), wrote the proofs, and wrote the first draft of the work. Sebastian and Patrick supervised this work, and made suggestions that improved the quality of the original draft.

Chapter 4 The idea for this chapter was developed jointly between me, Patrick, and Sebastian. The contents of this chapter were written by the author. The proof for Proposition 9 was generously provided to me by Horatio Boedihardjo by email. I implemented and ran the numerical experiments.

Chapter 5 The idea for this chapter was the author's. I implemented the methods, ran the numerical experiments, and wrote the first draft for this chapter. Patrick and Sebastian supervised this project and helped to finalise the published paper that resulted from this chapter's work.

Part III Part III of this thesis includes Chapter 6, which is derived from work I have done with Patrick Cannon and Sebastian M. Schmon of the Research Team at Improbable, and my PhD supervisor at Oxford, J. Doyne Farmer. Doyne helped to supervise and guide the direction of the project. My contributions are the following:

Chapter 6 I designed and conducted the experiments presented in this chapter. I also wrote the text, incorporating comments and feedback from Patrick and Sebastian that improved the clarity of the arguments we make in favour of the methods investigated in this chapter.

Part IV The entirety of the Epilogue was written by the author.

Appendices All appendices were written by the author.

Papers

As mentioned above, some of the work in this thesis is based on work either already published or released as preprints, working papers, or workshop papers. They are the following:

1. **Part I** incorporates material from the introductory sections of our preprint Dyer et al. (2021a),

   Approximate Bayesian Computation with Path Signatures
   Joel Dyer, Patrick Cannon, Sebastian M. Schmon
   arXiv:2106.12555v1, 2021

   our workshop paper Dyer et al. (2021b),

   Deep Signature Statistics for Likelihood-free Time-series Models
   Joel Dyer, Patrick Cannon, Sebastian M. Schmon
   ICML 2022 Time-series Workshop, 2021

   our published paper Dyer et al. (2022c),

   Amortised Likelihood-free Inference for Expensive Time-series Simulators with Signatured Ratio Estimation
   Joel Dyer, Patrick Cannon, Sebastian M. Schmon
   Artificial Intelligence in Statistics (*AISTATS*), 2022

   and our working paper Dyer et al. (2022a),

   Black-box Bayesian Inference for Economic Agent-based Models
   Joel Dyer, Patrick Cannon, J. Doyne Farmer, Sebastian M. Schmon
   INET Oxford Working Paper, 2022.

2. **Chapter 2** is loosely based on introductory sections from our published paper Dyer et al. (2022c) and our preprint Dyer et al. (2021a), both of which are already listed above.

3. **Chapter 3** is also derived from our preprint Dyer et al. (2021a), which is already listed above.

4. **Chapter 4** is derived from the workshop paper Dyer et al. (2021b), which is already listed above.

5. **Parts III and IV** are derived from our working paper Dyer et al. (2022a), which is already listed above, and our workshop paper Dyer et al. (2022b),

   Calibrating Agent-based Models to Microdata with Graph Neural Networks
   Joel Dyer, Patrick Cannon, J. Doyne Farmer, Sebastian M. Schmon
   ICML 2022 Workshop on Artificial Intelligence for Agent-based Modelling, 2022.

Other papers

During my DPhil, the paper Dyer and Kolic (2020) was also written and published in collaboration with Blas Kolic, a fellow DPhil student at INET Oxford. It covers a different topic to this thesis, and has therefore been omitted from this manuscript:

Public Risk Perception and Emotion on Twitter during the Covid-19 Pandemic,
Joel Dyer, Blas Kolic
Applied Network Science, 2020.

Additionally, after submitting this thesis, the workshop paper Dyer et al. (2022d) – which considers extensions discussed in Section 3.3.2 of this thesis – was accepted and presented at the NeurIPS 2022 Temporal Graph Learning Workshop:

Approximate Bayesian Computation for Panel Data with Signature Maximum Mean Discrepancies,
Joel Dyer, John Fitzgerald, Bastian Rieck, Sebastian M Schmon
NeurIPS Temporal Graph Learning Workshop, 2022.

Contents

List of Figures

Part I

Introduction

Chapter 1

Bayesian Statistical Inference[1]

To understand and make predictions about the world, human beings construct models – implicitly or explicitly – of their environments. Such models provide a simplified version of the real-world system of interest, capturing only those aspects of the world believed to be essential to reproducing the key features of the behaviour of that system, and are often quantitative and expressed mathematically as a series of equations. A mathematical approach towards modelling the world has a long history in the physical sciences, dating at least as far back as the 17th Century during which, for example, Johannes Kepler developed and published the eponymous laws of planetary motion using astronomical data collected while working with Tycho Brahe (Stephenson and Kepler, 1994). Nowadays, mathematical models are used and developed across fields ranging from economics (Kydland and Prescott, 1982; Korobow et al., 2007; Baptista et al., 2016) to biology (Turing, 1952; Baker et al., 2008; Christensen et al., 2015).

While mathematical models have proved invaluable to the progression of human understanding of the universe, they are by definition limited in the degree to which they can faithfully represent any real-world system of interest. Consequently, details that are important to the dynamics of the system under consideration are often omitted by necessity. For example, models of open systems – systems that are subjected to external influences – necessarily omit details regarding the behaviour of these external influences. However, even when the system being modelled lacks significant

[1]The literature review presented in this chapter is derived elements from the background sections of Dyer et al. (2021a), Dyer et al. (2021b), Dyer et al. (2022c), and Dyer et al. (2022a). The first three of these are joint work with Patrick Cannon and Sebastian M. Schmon, while the last is joint work with Patrick Cannon, J. Doyne Farmer, and Sebastian M. Schmon.

external influences and are to a good approximation "closed", any model of the system may still fail to endogenously capture the internal mechanisms that determine its behaviour with sufficient detail to accurately reproduce that behaviour.

In response to these limitations imposed on mathematical models, modellers frequently attempt to indirectly capture physical effects that are important to the behaviour of the system, but that cannot be captured in full detail by the model, by introducing an element of *randomness* or *stochasticity*. This inclusion of stochasticity may not itself represent a belief that the underlying process is inherently random, only that the processes that are not captured endogenously by an explicit deterministic component of the model are adequately described by one or more random variables.

Throughout this thesis, we will be concerned with such non-deterministic, probabilistic models. Probabilistic mathematical models may be viewed as their associated probability mass or density functions $p(\mathbf{x} \mid \boldsymbol{\theta})$ – also known as the *likelihood function* – for discrete or continuous data $\mathbf{x}$, respectively, where $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ are parameters describing the relationship between the model's constituent variables and $\boldsymbol{\Theta}$ is an arbitrary parameter space. We will consider only the case of $\boldsymbol{\Theta} \subset \mathbb{R}^q$ for some fixed (model-dependent) $q \in \mathbb{N}$ in this thesis. The non-determinism in such probabilistic models then manifests itself as a varying behaviour for identical inputs $\boldsymbol{\theta}$, such that the output of the model for a fixed value of $\boldsymbol{\theta}$ is a random variable distributed according to $p(\cdot \mid \boldsymbol{\theta})$.

In particular, we will be broadly concerned in this thesis with the problem of connecting arbitrary probabilistic models to empirical data $\mathbf{y}$: data that is generated by and collected from the real-world system under consideration. Making the connection between a probabilistic model $p(\cdot \mid \boldsymbol{\theta})$ and observed data $\mathbf{y}$ is usually achieved by estimating the model parameters $\boldsymbol{\theta}$ having observed $\mathbf{y}$: that is, by performing statistical parameter inference. This is a classical problem in statistics, with a number of canonical approaches to doing so. We describe these in Sections 1.1 and 1.2.

1.1 Frequentist inference

A dominant statistical inference paradigm, which relies on the conception of probabilities as long-run frequencies, is *frequentist* inference. Under this framework, it is assumed that a single, true value of $\boldsymbol{\theta}$ exists that gives rise to data $\mathbf{y}$. Inference for

$\boldsymbol{\theta}$ may then proceed by, for example, constructing an estimator $\hat{\boldsymbol{\theta}}(\mathbf{X})$ of $\boldsymbol{\theta}$, which is a function of the random output $\mathbf{X}$ of the model and returns a point estimate of $\boldsymbol{\theta}$ when data is observed. While any estimator can in principle be used for this purpose, an intuitively appealing estimator is the maximum likelihood estimate (MLE),

$$\hat{\boldsymbol{\theta}}(\mathbf{y}) := \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} p(\mathbf{y} \mid \boldsymbol{\theta}); \tag{1.1}$$

that is, the parameter that maximises the probability (density) of the observed data assuming the model $\{p(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$.

Frequentist parameter inference is, however, not limited to obtaining point estimates of the parameter of interest. For example, an alternative frequentist inference task is to perform a hypothesis test, which in general takes the form of constructing a test statistic $t(\mathbf{X}) \in \mathbb{R}^n$ and a critical region $C \subset \mathbb{R}^n$ for some $n \in \mathbb{N}$, on the basis of which we determine whether to (a) reject the null hypothesis $H_0 : \boldsymbol{\theta} \in \boldsymbol{\Theta}_0$ in favour of an alternative hypothesis $H_1 : \boldsymbol{\theta} \in \boldsymbol{\Theta}_1$ or (b) not, where $\boldsymbol{\Theta}_0 \cap \boldsymbol{\Theta}_1 = \phi$. This is done with the following rule: reject the null hypothesis if $t(\mathbf{X}) \in C$, otherwise do not reject the null hypothesis.

1.2 Bayesian inference

An alternative dominant inference paradigm to frequentist inference is Bayesian inference. In contrast to frequentist statistics, which is based on the notion that a single true value for $\boldsymbol{\theta}$ exists, Bayesian inference treats $\boldsymbol{\theta}$ as a random variable and instead considers *degrees of belief* regarding the values that $\boldsymbol{\theta}$ can appropriately assume. Inference proceeds by then updating one's degree of belief about appropriate values for $\boldsymbol{\theta}$ when new data is observed.

Central to this task of updating beliefs is the task of placing of probability distributions over $\boldsymbol{\Theta}$. Broadly speaking, Bayesian inference proceeds as follows:

1. specify one's model for the data-generating process, yielding a likelihood function $p(\mathbf{x} \mid \boldsymbol{\theta})$ which gives the probability (density) for any data $\mathbf{x}$ at each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$;

2. on the basis of all prior information and knowledge, encode one's initial beliefs about appropriate values for $\boldsymbol{\theta}$ by constructing a probability distribution $\pi : \boldsymbol{\Theta} \to \mathbb{R}_{\geq 0}$ over $\boldsymbol{\theta}$. This distribution is termed the *prior* distribution over $\boldsymbol{\Theta}$;

3. update one's belief distribution on the basis of observed data $\mathbf{y}$ according to *Bayes's theorem*,

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta})}{p(\mathbf{y})}\pi(\boldsymbol{\theta}), \tag{1.2}$$

where

$$p(\mathbf{y}) = \int_{\boldsymbol{\Theta}} p(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \tag{1.3}$$

is termed the *prior predictive* distribution, which is the marginal distribution over data $\mathbf{y}$. The probability density function on the left-hand side of Equation (1.2) is termed the *posterior* distribution, and captures one's *updated beliefs* about appropriate values for $\boldsymbol{\theta}$ having observed new data $\mathbf{y}$, while the ratio $p(\mathbf{y} \mid \boldsymbol{\theta})/p(\mathbf{y})$ on the right-hand side is often termed the *likelihood-to-evidence ratio*.

The parameter posterior distribution, given by the correct application of Bayes's theorem, then captures all inferences one can make about $\boldsymbol{\theta}$ given data $\mathbf{y}$. Such inferences may include, for example: the posterior mean of a function $g : \boldsymbol{\Theta} \to \mathbb{R}$,

$$\mathbb{E}_{\pi(\boldsymbol{\theta}|\mathbf{y})}\left[g(\boldsymbol{\theta}) \mid \mathbf{y}\right] = \int_{\boldsymbol{\Theta}} g(\boldsymbol{\theta})\pi(\boldsymbol{\theta} \mid \mathbf{y})\mathrm{d}\boldsymbol{\theta}; \tag{1.4}$$

predictions about the future of the system via the *posterior predictive distribution*,

$$p(\mathbf{y}' \mid \mathbf{y}) := \mathbb{E}_{\pi(\boldsymbol{\theta}|\mathbf{y})}\left[p(\mathbf{y}' \mid \boldsymbol{\theta})\right] = \int_{\boldsymbol{\Theta}} p(\mathbf{y}' \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta} \mid \mathbf{y})\mathrm{d}\boldsymbol{\theta}; \tag{1.5}$$

or the maximum *a posteriori* (MAP) estimate,

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} := \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \pi(\boldsymbol{\theta} \mid \mathbf{y}). \tag{1.6}$$

The decision between assuming the frequentist and Bayesian statistical frameworks for inferential tasks is a long-standing debate and is in large part a matter of philosophy (Bayarri and Berger, 2004). However, proponents of the Bayesian framework offer arguments in favour of Bayesian inference such as the fact that it naturally provides uncertainty quantification via the posterior density, that it permits the incorporation of prior information and knowledge via the prior distribution, and that it facilitates more intuitive statements about the inferences one can draw, which has advantages when communicating technical results to non-technical decision-makers in the real world. There is furthermore evidence that model averaging via the posterior density provides better predictions and averts difficulties introduced by overconfidence (Aitchison, 1975). As a result, our primary focus in this thesis will be on the Bayesian framework.

1.2.1 Approximating intractable posterior distributions

We briefly note that most posterior densities have no closed form expression for arbitrary (and in some cases even simple) probabilistic models and priors, even when the likelihood and prior themselves are available in closed form. In such cases, it is possible to approximate the true posterior, for example by simulation: sampling from the posterior distribution, such that the frequency with which values occur reflect the posterior densities of those values. This may be performed in various ways, including via Markov chain Monte Carlo (MCMC) e.g. Metropolis-Hastings (Hastings, 1970b) and slice sampling (Neal, 2003), or for example via sequential Monte Carlo (Doucet et al., 2001).

1.3 Summary statistics

For certain probabilistic models – and for certain approaches to approximating statistical parameter inference for many probabilistic models, as we will discuss later in this thesis – the data $\mathbf{y}$ does not or cannot appear in its entirety. Instead, a summary statistic $\mathbf{s}(\mathbf{y})$ of $\mathbf{y}$ often appears, and is the lens through which the inference procedure views the data. As an example, consider an experiment in which a coin with a fixed and unknown probability θ of Heads is flipped n times. By letting $\mathbf{X}_i$ be the indicator function for the event $\{\text{coin toss } i \text{ yields Heads}\}$, $i = 1, \ldots, n$, the probability of the data is

$$p(\mathbf{X}_1 = \mathbf{x}_1, \ldots, \mathbf{X}_n = \mathbf{x}_n \mid \theta) = \prod_{i=1}^{n} \theta^{\mathbf{x}_i} (1-\theta)^{1-\mathbf{x}_i} \tag{1.7}$$

$$= \theta^{\sum_{i=1}^{n} \mathbf{x}_i} (1-\theta)^{n-\sum_{i=1}^{n} \mathbf{x}_i}. \tag{1.8}$$

In this instance, we see that the model only interacts with the data via the statistic

$$\mathbf{s}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \mathbf{x}_i, \tag{1.9}$$

such that all inferences regarding θ are made only on the basis of the value of $\sum_{i=1}^{n} \mathbf{x}_i$.

The example in Equations (1.7)-(1.9) further serves as an example of a special class of summary statistics. Intuitively, and as a result of the data-processing inequality that states that a (deterministic or probabilistic) function of a random variable $\mathbf{X}$ can

contain no more information about $\mathbf{X}$ than $\mathbf{X}$ itself, a summary statistic $\mathbf{s}(\mathbf{y})$ contains no additional information about the data than the data itself, and may contain less. Summary statistics that do not incur any loss of information that is relevant to the parameter $\boldsymbol{\theta}$ being inferred, and that are in this sense completely informative about $\mathbf{y}$ with respect to the given model $\{p(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, are referred to as *sufficient statistics*. More formally:

Definition 1 (Kolmogorov (1942))**.** *A summary statistic* $\mathbf{s}(\mathbf{x})$ *of* $\mathbf{x}$ *is said to be* sufficient *with respect to a model* $p(\cdot \mid \boldsymbol{\theta})$ *if for every prior distribution* $\pi(\boldsymbol{\theta})$ *we have that*

$$\pi(\boldsymbol{\theta} \mid \mathbf{s}(\mathbf{x})) = \pi(\boldsymbol{\theta} \mid \mathbf{x}). \tag{1.10}$$

That is, all inferences drawn by conditioning on $\mathbf{s}(\mathbf{x})$ *are identical to those drawn by conditioning on* $\mathbf{x}$*.*

The above definition is sometimes referred to as "Bayesian sufficiency", in contrast to the following notion of classical, frequentist sufficiency:

Definition 2 (Fisher and Russell (1922))**.** *A summary statistic* $\mathbf{s}(\mathbf{x})$ *of* $\mathbf{x}$ *is said to be* sufficient *with respect to a model* $p(\cdot \mid \boldsymbol{\theta})$ *if*

$$p(\mathbf{y} \mid \mathbf{s}(\mathbf{y}), \boldsymbol{\theta}) = p(\mathbf{y} \mid \mathbf{s}(\mathbf{y})). \tag{1.11}$$

That is, $\mathbf{y}$ *and* $\boldsymbol{\theta}$ *are conditionally independent given* $\mathbf{s}(\mathbf{y})$*.*

Classical sufficiency implies Bayesian sufficiency, although the converse is not always true and is more complicated (Blackwell and Ramamoorthi, 1982). A result known as the Fisher-Neyman factorisation theorem enables verification of the sufficiency of a summary statistic by providing a necessary and sufficient condition for sufficiency:

Proposition 1 (Neyman (1935))**.** *A vector-valued statistic* $\mathbf{s}(\mathbf{X}_1, \ldots, \mathbf{X}_n)$ *is jointly sufficient for* $\boldsymbol{\theta}$ *iff the following factorization holds:*

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid \boldsymbol{\theta}) = g\left(\boldsymbol{\theta}, \mathbf{s}(\mathbf{x}_1, \ldots, \mathbf{x}_n)\right) h(\mathbf{x}_1, \ldots, \mathbf{x}_n), \tag{1.12}$$

where $g(\cdot, \cdot)$ *and* $h(\cdot)$ *are non-negative functions,* $\boldsymbol{\theta}$ *does not appear in* h*, and the* $\mathbf{x}_i$ *only appear in* g *via* $\mathbf{s}$*.*

We therefore see that the summary statistic given by Equation (1.9) is indeed sufficient with respect to the model in Equation (1.7), since Equation (1.7) exhibits the required factorisation with $h(\mathbf{x}_1, \ldots, \mathbf{x}_n) = 1$.

A corollary of Proposition 1 is that an injective function of a sufficient statistic is itself sufficient. While it is a useful guiding ideal, a low-dimensional sufficient statistic is known to be unattainable in general, due to the Pitman-Koopman-Darmois Theorem:

Proposition 2 (Pitman (1936); Koopman (1936); Darmois (1935))**.** *Of all families of probability distributions for which the domain of the distribution does not vary with the parameter that is to be estimated, it is only exponential families that permit a sufficient statistic whose dimension remains bounded as the size of the sample increases.*

A consequence of this theorem is that arbitrary probabilistic models do not permit a sufficient statistic whose size does not grow with the number of data points observed. This has important implications for the task of Bayesian inference in practice, and specifically for approximations thereof as we will discuss throughout later chapters.

1.4 Simulation models & likelihood-free Bayesian inference

In the previous sections, we discussed the basic problem of Bayesian parameter inference as that of evaluating the posterior distribution over parameter values after observing a real-world dataset $\mathbf{y}$. This is obtained by application of Bayes's theorem, given in Equation 1.2. We further discussed the fact that there usually exists no closed-form expression for the posterior distribution for arbitrary – but also even for simple – choices of prior distribution $\pi(\boldsymbol{\theta})$ and model $p(\mathbf{y} \mid \boldsymbol{\theta})$, necessitating the use and development of procedures such as MCMC for approximating the posterior distribution.

In this section, we will expand on the above by introducing in more detail the topic of this thesis, and by discussing the relationship between this topic and the contents of the previous sections. We will begin by discussing *simulation models*, which are becoming increasingly popular across domains in the natural and social sciences – in addition to within more applied fields such as engineering and with significant uptake by policy-making institutions – as a tool for understanding and predicting

the behaviour of systems of interests. We will then discuss one barrier to their use in real-world decision-making scenarios, which will establish the topic of this thesis. Finally, we will conclude this chapter with an outline of the remainder of this thesis.

1.4.1 Simulation models

The modern digital age, and the cheap and widely available computational resources of the 21st Century, presents us with a plethora of unprecedented opportunities in both applied and scientific domains. One such opportunity that concerns us here is the ease with which simulation models can be developed and used to study complex systems. Simulation models are models that are defined implicitly by an underlying computer program, the purpose of which is often to directly model the mechanisms and processes determining the interactions between and behaviours of the constituent parts of the system of interest. For example, simulation models in biology may be used to directly model the processes according to which cells in a tumour divide, the ultimate goal of which being to understand the development of the tumour or progression of the underlying disease. A further example is a simulation model in which the behavioural and interaction rules for a multitude of heterogeneous agents in a social system are implemented, with the ultimate goal of understanding how opinions, beliefs, or culture evolves within a population of individuals. Simulation models are however not restricted to these scientific domains, and are applied and employed in various fields ranging from economics (Baptista et al., 2016) and epidemiology (Kerr et al., 2021) – for example via the use of agent-based models (Bonabeau, 2002) – to cosmology, for example in order to model supernovae (Alsing et al., 2018).

Simulation as a modelling paradigm has significant advantages. One such benefit is that simulating affords the modeller greater flexibility than has historically been afforded mathematical modellers: when a simulation study of the model is possible, analytic tractability is no longer a restriction imposed on the form of the model. Consequently, the modeller is free to specify the model as they desire, and is not forced into assuming simpler or more convenient mathematical expressions in the interest of preserving analytic tractability. Furthermore, modellers are able to study complex systems and emergent phenomena via *bottom-up* simulation: the modeller is able to implement the rules governing the behaviour of the system's microscopic constituents, before simply simulating and observing – rather than undertaking the arduous and often impossible task of reasoning about – the resultant emergent macroscopic behaviour of the system. Finally, simulation in the social sciences and similar fields, in

which there is significant interaction between and heterogeneity across the constituent parts (e.g. households, firms etc.) of the system of interest, is beneficial in that certain classes of simulation models, such as agent-based models (Bonabeau, 2002), more naturally permit the incorporation of these interactions and heterogeneities than more conventional modelling paradigms (Farmer and Foley, 2009). Such models thus present a significant advantage over alternative modelling tools when these effects are essential to the behaviour of the system.

1.4.2 Likelihood-free inference & time-series simulators

While simulation models have numerous advantages, they additionally pose challenges to their use in real-world decision-making. The problem that we consider in this thesis is that stochastic simulation models do not generally possess a tractable likelihood function, in the sense that it is usually difficult, for example due to computational expense, to evaluate $p(\mathbf{y} \mid \boldsymbol{\theta})$ for some or all $\boldsymbol{\theta}$. It is not difficult to imagine such scenarios. For example, the model of interest may contain a high-dimensional set of latent variables $\mathbf{z}$, such that evaluation of the model likelihood for observed data $\mathbf{y}$ amounts to evaluating the following integral:

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \int_{\mathcal{Z}} p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) \mathrm{d}\mathbf{z}, \tag{1.13}$$

where $\mathcal{Z}$ is the domain of $\mathbf{z}$. For high-dimensional $\mathbf{z}$, obtaining a closed-form expression for Equation (1.13) will in general be impossible, and evaluating it numerically can furthermore be an intractable or otherwise undesirable computational task. A further, more practical constraint on the tractability of simulation models' likelihood functions is the fact that they are often in practice *black-box simulators*, in the sense that one does not have access to the simulator's internal workings. This can result from a number of causes, for example that the simulator itself is written in a difficult-to-read low-level programming language for computational reasons, or that the simulation code is proprietary or confidential, such that there exist restrictions on access to its code.

The consequence of this feature of simulation models that we consider in this thesis is that the problem of statistical parameter inference, discussed in the previous chapter, is made particularly complicated. This is due to the fact that many of the most popular traditional approaches to parameter inference – such as the MLE and the

use of Bayes's theorem in Equations (1.1) and (1.2), respectively – make explicit use of the likelihood function. This raises the question of whether there exists good procedures for approximating such traditional inference methods that account for the intractability of simulation models' likelihood functions.

This challenge has inspired significant work within the computational statistics community over the past two decades, from which various procedures for approximating inference for intractable likelihood models have emerged. These are often termed likelihood-free inference (LFI) or simulation-based inference (SBI) procedures (we will use both interchangeably), and it is this class of inference procedures that concern us here. In the following section, we provide an overview of the different streams of work that comprise the state-of-the-art in simulation-based/likelihood-free parameter inference. As we will discuss later in this Chapter, one of the key challenges faced within the likelihood-free inference literature that we seek to address with our work is making different simulation-based inference procedures compatible with time-series data of various kinds. Our discussion will be somewhat focused on this aspect of the literature on likelihood-free inference for this reason. It will also be focused on the literature on simulation-based inference for models in the social sciences, since this is another sub-focus of this thesis.

1.5 Literature review & key challenges

1.5.1 Approximate Bayesian computation

We will first recapitulate some standard approaches to approximate Bayesian computation (ABC) with an emphasis on time series data. Let $\mathcal{X}^n$ be the space of all length n sequences taking values in $\mathcal{X}$ and suppose we have time series data $\mathbf{y} = (\mathbf{y}_{t_1}, \mathbf{y}_{t_2}, \ldots, \mathbf{y}_{t_n}) \in \mathcal{X}^n$, observed at real times $0 = t_1 < t_2 < \ldots < t_n = T$, and assumed to have been drawn from the generative model with density $p(\mathbf{y} \mid \boldsymbol{\theta})$ parameterised by $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p) \in \boldsymbol{\Theta} \subseteq \mathbb{R}^p$. Given a prior distribution $\pi(\boldsymbol{\theta})$ on $\boldsymbol{\Theta}$, the central object in Bayesian inference is the posterior distribution

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto p(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{1.14}$$

For simulation models, the likelihood function $p(\mathbf{y} \mid \boldsymbol{\theta})$ is commonly intractable, in the sense that it cannot be evaluated point-wise, making infeasible standard Bayesian approaches to posterior inference such as Markov chain Monte Carlo (MCMC).

In such scenarios, an established alternative is offered by ABC (Tavaré et al., 1997; Pritchard et al., 1999; Beaumont et al., 2002) which allows the user to approximate the true posterior (1.2) using only forward samples from the simulator. Broadly, the user is required to specify summary statistics $\mathbf{s} : \mathcal{X}^n \to \mathcal{S}$ – where typically $S \subseteq \mathbb{R}^k$, $k \in \mathbb{N}$, or $\mathcal{X}^n$ when $\mathbf{s}$ is simply the identity map – after which the true likelihood function is approximated as

$$\hat{p}_{\boldsymbol{\theta}}(\mathbf{y}) = \int K_\varepsilon \left[\mathbf{s}\left(\mathbf{y}\right), \mathbf{s}\left(\mathbf{x}\right)\right] \cdot p(\mathbf{x} \mid \boldsymbol{\theta}) \, \mathrm{d}\mathbf{x}, \tag{1.15}$$

where $K_\varepsilon(\cdot, \cdot)$ is a kernel function with parameters ε. The resulting ABC posterior is then given by

$$\pi_{\text{ABC}}\left(\boldsymbol{\theta} \mid \mathbf{y}\right) \propto \hat{p}_{\boldsymbol{\theta}}(\mathbf{y}) \cdot \pi\left(\boldsymbol{\theta}\right), \tag{1.16}$$

which – when $\epsilon \in \mathbb{R}$ is simply a bandwidth parameter – is consistent as $\varepsilon \to 0$ if the employed summary statistic is sufficient, since as $\varepsilon \to 0$, $K_\varepsilon \left[\mathbf{s}\left(\mathbf{y}\right), \mathbf{s}\left(\mathbf{x}\right)\right] \to \delta_{\mathbf{y}}(\mathbf{x})$, and so the right hand side of Equation (1.15) approaches $p(\mathbf{y} \mid \boldsymbol{\theta})$. Additionally, extending upon the concept of generalized Bayesian inference (Bissiri et al., 2016; Knoblauch et al., 2019), Schmon et al. (2020) note that ABC can be seen as a generalized Bayesian method targeting the posterior

$$\pi_{\text{GBI}}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \int e^{-w \cdot \ell(\mathbf{y}; \mathbf{x})} p(\mathbf{x} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) \mathrm{d}\mathbf{x} \tag{1.17}$$

for an arbitrary loss function $\ell(\mathbf{y}; \mathbf{x})$ that captures the discrepancy between observation $\mathbf{y}$ and simulation $\mathbf{x}$, and some weight hyperparameter $w \in \mathbb{R}$.

The approach as presented above leaves open a plethora of possible choices for $\mathbf{s}$ and $K_\varepsilon(\cdot, \cdot)$—or, more generally, the loss function ℓ and hyperparameter w—which has sparked great interest in the choice of those values in different scenarios, the complete enumeration of which is beyond the scope of this overview. However, we summarise here some of the most common approaches.

Rejection ABC The standard rejection ABC (REJ-ABC) algorithm corresponds to choosing a uniform kernel $K_\varepsilon(\cdot, \cdot) \propto \mathbb{1}\left[\rho\{\cdot, \cdot\} \leq \varepsilon\right]$, where $\rho : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ is some distance function. That is, parameter values $\boldsymbol{\theta}$ are independently drawn from the prior and are retained as samples from an approximate posterior according to whether the distance between $\mathbf{s}(\mathbf{y})$ and $\mathbf{s}(\mathbf{x})$ falls at or below a threshold ε. The choice of threshold ε is left to the experimenter, and for example may be determined in advance of the inference procedure, or chosen after simulation time such that a certain proportion of the total simulation budget is retained.

Semi-automatic ABC Fearnhead and Prangle (2012) propose a method for automatically generating low-dimensional summary statistics by reducing a larger candidate set of summaries, referred to as semi-automatic ABC (SA-ABC). Their approach may be summarised as follows: Given a set of N training data points $\left(\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}\right) \sim p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$, $i = 1, \ldots, N$, and a candidate vector $\mathbf{g}(\cdot)$ of J summary statistics, the method proceeds by performing vector-valued linear regression from $\mathbf{g}(\mathbf{x}^{(i)})$ to $\boldsymbol{\theta}^{(i)}$, producing a matrix A of coefficients. The summaries $\mathbf{s}$ are then taken to be the output of this regression, i.e. $\mathbf{s}(\mathbf{x}^{(i)}) = A\mathbf{g}(\mathbf{x}^{(i)})$. The motivation for this is that, under certain limits and a quadratic loss, the ABC summary statistics that yield the minimum loss between the true parameter value and point estimates from the ABC posterior can be shown to be the posterior mean $\mathbb{E}[\boldsymbol{\theta} \mid \mathbf{y}]$. A drawback of this method, however, is that it requires the construction of an initial set of candidate summaries, which would need to be informative. Other approaches in this vein include that of Nakagome et al. (2013), in which the authors propose the use of SA-ABC using kernel ridge regression, to exploit the nonlinearities induced by kernel methods in this regression task. We provide a more detailed review on learning summary statistics in this fashion in Section 1.5.5.

K2-ABC Park et al. (2016) propose double kernel ABC (K2-ABC), an ABC method that bypasses the problem of constructing summary statistics for *iid* data by using the maximum mean discrepancy (MMD) between (a) the simulator's distribution $f(\cdot \mid \boldsymbol{\theta})$, where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \sim p(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} f(\mathbf{x}_i \mid \boldsymbol{\theta})$, and (b) the true density f^* giving rise to the *iid* observations comprising $\mathbf{y}$, respectively. That is, with a suitable kernel k, the discrepancy between the simulation output $\mathbf{x}$ and observation $\mathbf{y}$ is then taken to be the squared MMD

$$\text{MMD}^2 = \|\mathbb{E}_{\mathbf{z} \sim f(\cdot \mid \boldsymbol{\theta})}[k(\mathbf{z}, \cdot)] - \mathbb{E}_{\mathbf{z}' \sim f^*}[k(\mathbf{z}', \cdot)]\|_{\mathcal{H}}^2, \tag{1.18}$$

where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) associated with k. In this way, the choice of summary statistics (e.g. as required in SA-ABC) can be seen as being replaced by the choice of kernel k. For time series data, the authors suggest that the dependency structure can be ignored, and that the observation $\{\mathbf{y}_i\}_{i=1}^{n}$ and simulation output $\{\mathbf{x}_i\}_{i=1}^{m}$ can still be treated as *iid* data from the marginal densities $f(\cdot \mid \boldsymbol{\theta}) := f_{\boldsymbol{\theta}}$ and f^*, respectively. An unbiased estimate of the MMD, under this

assumption, can thus be obtained as

$$\widehat{\mathrm{MMD}}^2(f_{\boldsymbol{\theta}}, f^*) = \frac{1}{m(m-1)} \sum_{i \neq j} k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\mathbf{y}_i, \mathbf{y}_j) - \frac{2}{nm} \sum_{\substack{i=1,\dots,m \\ j=1,\dots,n}} k(\mathbf{x}_i, \mathbf{y}_j). \tag{1.19}$$

Wasserstein ABC Bernton et al. (2019) propose a further method for measuring the discrepancy between observations and simulated data that circumvents the problem of manually constructing summary statistics. The approach uses as its measure of discrepancy the p-Wasserstein distance between the empirical distribution of observations $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$, and simulated data $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$, with $\mathbf{y}_i, \mathbf{x}_j \in \mathbb{R}^d$. That is, the distance ρ is taken to be

$$\mathcal{W}_p(\mathbf{y}, \mathbf{x})^p = \inf_{\gamma \in \Gamma_{n,m}} \sum_{i=1}^{n} \sum_{j=1}^{m} \rho_0(\mathbf{y}_i, \mathbf{x}_j)^p \gamma_{ij} \tag{1.20}$$

where ρ_0 is a distance on $\mathbb{R}$ and $\Gamma_{n,m}$ is the set of $n \times m$ matrices with non-negative entries, columns summing to m^{-1}, and rows summing to n^{-1}. The authors propose to use $p = 1$, in order to make a minimal number of assumptions on the existence of moments of the data-generating process.

A number of solutions are proposed in Bernton et al. (2019) to account for the dependency structure inherent in time series data. The first strategy discussed is the use of *curve matching*, in which a time augmentation $\mathbf{y}_{t_i} \mapsto (t_i, \mathbf{y}_{t_i})$ is applied to the data, and the following ground distance between elements of the sequence used:

$$\rho_0\left((t_i, \mathbf{y}_{t_i}), (t_j, \mathbf{x}_{t_j}); \lambda\right) = \|\mathbf{y}_{t_i} - \mathbf{x}_{t_j}\| + \lambda |t_i - t_j| \tag{1.21}$$

where $\lambda > 0$ is a free parameter that interpolates the distance in (1.20) between the sum of Euclidean distances $\sum_i \|\mathbf{y}_{t_i} - \mathbf{x}_{t_i}\|$ and the Wasserstein distance between the empirical marginal distributions of $\mathbf{y}$ and $\mathbf{x}$. A heuristic for tuning λ is offered only for the case of univariate $\mathbf{y}$ and $\mathbf{x}$.

A second strategy employs *reconstructions*, where the data are transformed to generate empirical distributions that allow for easier identifiability of parameters. Two types of reconstructions are considered: delay reconstructions, which is a common technique for reconstructing phase spaces in dynamical systems theory that involves

considering lagged sequences of observations from the data; and residual reconstructions, in which the data is transformed according to the structure of the generative model such that they become *iid* observations, for example by considering $\epsilon_t = (\mathbf{x}_t - a\,\mathbf{x}_{t-1})/\sigma$ in the case of a centered AR(1) model with parameter $\boldsymbol{\theta} = (a, \sigma)$. However, it is undesirable to rely on such methods. For the case of delay reconstructions, properly estimating the lag parameters is key to its success (Fraser and Swinney, 1986) and obtaining reliable estimates remains a significant challenge (Bradley and Kantz, 2015). This is likely to be exacerbated in likelihood-free inference (LFI) settings, in which time series are stochastic and are often short, due to computational expense. Delay reconstructions will then also further reduce this length of the data, which can be costly to the quality of the inference procedure. Furthermore, the often complicated or unknown internal mechanisms of complex simulation models typically do not allow for a simple transformation of the output into *iid* data, limiting the applicability of this approach in LFI settings.

1.5.2 Approximate Bayesian computation in the social sciences

Bayesian parameter inference for simulation models in the social sciences – such as agent-based models (ABMs) in economics – has gained popularity with, for example, the work of Grazzini et al. (2017) and Platt (2021) in economics. Below, we outline these approaches, and show that each method can be comprehensively subsumed under "ABC".

1.5.2.1 Parametric density estimation

The simplest approach discussed by Grazzini et al. (2017) involves assuming that the simulation model has entered a statistical equilibrium, such that the observations $\mathbf{y}_t \in \mathbb{R}^d$ in the time-series $\mathbf{y} := (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T) \in \mathbb{R}^{d \times T}$ are fluctuations around some stationary value m^*:

$$\mathbf{y}_t = m^* + \varepsilon_t, \tag{1.22}$$

where the ε_t are *iid* noise terms with density g_ϵ and parameters ϵ. Although more complex distributions are available, g_ϵ may for example be a zero-mean Gaussian distribution, and ϵ the elements of the covariance matrix.

Under these assumptions, recalling that the elements of the time-series are assumed to be independent, the parameters m^* and ϵ can be estimated by the means and covariances of the elements of the time-series to give $\hat{m}^*$ and $\hat{\epsilon}$. More precisely, first fixing $\boldsymbol{\theta}$ and drawing a sample $\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$, one can calculate the estimates $\hat{m}^*(\mathbf{x})$ and $\hat{\epsilon}(\mathbf{x})$ using, for example, maximum likelihood estimation. Adopting as the measure of distance the probability density of the true data being observed under this approximated process yields the choice of kernel

$$K_\epsilon(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^{T} g_{\hat{\epsilon}(\mathbf{x})}\left(\mathbf{y}_t - \hat{m}^*(\mathbf{x})\right). \tag{1.23}$$

Finally by taking $R \geq 1$ *iid* simulated datasets[2] $\mathbf{x}^{(r)} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$, $r = 1, \ldots, R$, and calculating their associated estimators $\hat{m}^*\left(\mathbf{x}^{(r)}\right)$ and $\hat{\epsilon}\left(\mathbf{x}^{(r)}\right)$, the likelihood at $\boldsymbol{\theta}$, i.e. Equation (1.15), is approximated with the Monte Carlo average

$$\hat{p}_{\boldsymbol{\theta}}(\mathbf{y}) \approx \frac{1}{R} \sum_{r=1}^{R} \prod_{t=1}^{T} g_{\hat{\epsilon}\left(\mathbf{x}^{(r)}\right)}\left(\mathbf{y}_t - \hat{m}^*\left(\mathbf{x}^{(r)}\right)\right). \tag{1.24}$$

Alternatively, the R simulations may be pooled to generate single estimates $\hat{m}^*\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(R)}\right)$ and $\hat{\epsilon}\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(R)}\right)$ at $\boldsymbol{\theta}$. This is closely related to *synthetic likelihood* approaches (Wood, 2010; Price et al., 2018) which use similar Gaussian distributions, but usually rely on summary statistics.

In either case, the resulting approximate likelihood $\hat{p}_{\boldsymbol{\theta}}$ from Equation (1.24) can then be used downstream for parameter estimation, either directly via e.g. maximum likelihood or through the corresponding approximate posterior, simulated e.g. through MCMC. This approach suffers from clear limitations, however. Firstly, it treats the data points in the observed time-series as independent – that is, as lacking a natural ordering – which destroys important information when the observed $\mathbf{y}$ and simulated $\mathbf{x}$ are time-series. Secondly, choosing an appropriate and sufficiently flexible family of parametric densities g_ϵ to construct the likelihood approximation is non-trivial, with poor choices leading to erroneous Bayesian inference.

1.5.2.2 Non-parametric density estimation

An alternative approach, which partially addresses the second limitation described in Section 1.5.2.1, is to forgo the assumption of a parametric family of densities

[2]While R can be chosen to be any natural number, it is usually efficient to take $R = 1$ when averages of estimators are concerned and run the Markov chain for an accordingly higher number of iterations (Bornn et al., 2017; Sherlock et al., 2017).

and instead use a non-parametric method for density estimation. Grazzini et al. (2017) describe the use of kernel density estimation (KDE) for this purpose. Here, the data points in the time-series are once again assumed to be independent and fluctuating about some stationary value m^* as in the parametric approach described above. Then, an estimate of the likelihood function is obtained by applying KDE to $R \geq 1$ *iid* simulations of length S, $\mathbf{x}^{(r)} := (\mathbf{x}_1^{(r)}, \ldots, \mathbf{x}_S^{(r)}) \sim p(\mathbf{x} \mid \boldsymbol{\theta})$, $r = 1, \ldots, R$, providing an unbiased estimate of the approximated likelihood function in Equation (1.15) as

$$\hat{p}_{\boldsymbol{\theta}}(\mathbf{y}) \approx \frac{1}{R} \sum_{r=1}^{R} K_\epsilon\left(\mathbf{y}, \mathbf{x}^{(r)}\right) := \frac{1}{R} \sum_{r=1}^{R} \prod_{t=1}^{T} \hat{p}_\epsilon\left(\mathbf{y}_t \mid \boldsymbol{\theta}, \mathbf{x}^{(r)}\right), \tag{1.25}$$

where $\hat{p}_\epsilon\left(\mathbf{y}_t \mid \boldsymbol{\theta}, \mathbf{x}^{(r)}\right)$ is the estimate of the conditional density $p\left(\mathbf{y}_t \mid \boldsymbol{\theta}, \mathbf{x}^{(r)}\right)$ obtained via KDE:

$$\hat{p}_\epsilon(\mathbf{y}_t \mid \boldsymbol{\theta}, \mathbf{x}) = \frac{1}{S} \sum_{s=1}^{S} \kappa_\epsilon\left(\mathbf{y}_t - \mathbf{x}_s\right), \tag{1.26}$$

for some probability kernel κ_ϵ with bandwidth parameter(s) ϵ. In particular, the authors employ a Gaussian kernel with bandwidth chosen using Silverman's method (Silverman, 1986), such that

$$\kappa_\epsilon\left(\mathbf{y}_t - \mathbf{x}_s\right) = \frac{1}{\epsilon} \kappa\left(\frac{\|\mathbf{y}_t - \mathbf{x}_s\|_2}{\epsilon}\right). \tag{1.27}$$

Alternatively, as in Section 1.5.2.1, the R simulations may be pooled and a single KDE model fit to the combined dataset to obtain $\hat{p}(\mathbf{y} \mid \boldsymbol{\theta})$. While these non-parametric approaches to density estimation are arguably more flexible than the assumption of a parametric family, they are well known to suffer from the *curse of dimensionality*, limiting their applicability to low dimensional data only, even under the unrealistic assumption that the $\mathbf{y}_t$ are assumed independent.

1.5.2.3 Neural likelihood estimation methods in the social sciences

As an alternative to kernel density estimation, neural density estimators have previously been employed to perform Bayesian estimation of ABMs in the social sciences. Platt (2021) presents a method which diverges from the approaches of Grazzini et al. (2017) in two main regards. Firstly, in contrast to the previous independence assumption, it assumes an autoregressive structure for the time-series model to better capture temporal correlations in the data. In particular, the approach assumes a time-series model that is Markov of order L, that is, $p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \boldsymbol{\theta}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-L:t-1}, \boldsymbol{\theta})$, where

$\mathbf{x}_{1:t} = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$. In words, the distribution of the state at time t depends only on the previous L states (and $\boldsymbol{\theta}$). In light of this assumption, the full likelihood can be factorised as

$$p_{\boldsymbol{\theta}}(\mathbf{y}) = p(\mathbf{y}_{1:L} \mid \boldsymbol{\theta}) \prod_{t=L+1}^{T} p(\mathbf{y}_t \mid \mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-L}, \boldsymbol{\theta}). \tag{1.28}$$

Secondly, as a consequence of the autoregressive model structure, the method employs a *conditional* density estimator q_ϕ to approximate the transition density function $p(\mathbf{x}_t \mid \mathbf{x}_{t-L:t-1}, \boldsymbol{\theta})$. The resulting likelihood approximation can be written

$$\hat{p}_{\boldsymbol{\theta}}(\mathbf{y}) = \prod_{t=L+1}^{T} q_{\phi(\boldsymbol{\theta})}(\mathbf{y}_t \mid \mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-L}), \tag{1.29}$$

where the contribution of the first L terms is ignored. A mixture density network (Bishop, 1994) assumes the role of the conditional density estimator $q_{\phi(\boldsymbol{\theta})}$ with parameters $\phi(\boldsymbol{\theta})$ trained for each $\boldsymbol{\theta}$. Denoting the L states preceding $\mathbf{y}_t$ as $\mathbf{y}_{<t}$ for brevity, the particular form of the mixture density network is

$$q_\phi(\mathbf{y}_t \mid \mathbf{y}_{<t}) = \sum_{k=1}^{K} \alpha_k(\mathbf{y}_{<t}) \mathcal{N}(\mathbf{y}_t \mid \mu_k(\mathbf{y}_{<t}), \Sigma_k(\mathbf{y}_{<t})).$$

where α_k are the mixing coefficients, and μ_k, Σ_k are the mean and covariance matrix respectively, all of which are parameterised by neural networks and trained via maximum likelihood on R *iid* model samples $\mathbf{x}^{(r)} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$, $r = 1, \ldots, R$. The resulting likelihood estimate can then once again be used also with Algorithm 1, for example, to target the associated approximate posterior distribution.

While the model structure described above accounts for the sequential nature of simulations generated by time-series models such as ABMs to some extent, the computational burden associated with the simulation of sufficient training data and the training of a new conditional density estimator at each $\boldsymbol{\theta}$ renders it largely infeasible when the simulation model is expensive to run.

1.5.2.4 Unifying the approaches

In Algorithm 1, we show how the approximate posterior (1.16), with any of the possibilities for $\hat{p}_{\boldsymbol{\theta}}(\cdot)$ discussed above, may be targeted using a variant of the popular Metropolis-Hastings (MH) algorithm (e.g. Hastings, 1970a). If the likelihood estimate

Algorithm 1: Metropolis-Hastings sampling scheme for ABC

Input: Prior distribution $\pi(\cdot)$, observation $\mathbf{y}$, proposal distribution $q(\cdot \mid \boldsymbol{\theta})$, initial value $\boldsymbol{\theta}_0$, number of iterations n;
Result: Empirical posterior $\sum_{i=1}^{n} \delta_{\boldsymbol{\theta}_i}$
for $r = 1, \ldots, R$ **do**
  Simulate $\mathbf{x}^{(r)} \sim p(\cdot \mid \boldsymbol{\theta}_0)$;
end
for $i = 1, \ldots, n$ **do**
  Sample $\boldsymbol{\theta} \sim q(\cdot \mid \boldsymbol{\theta}_{i-1})$;
  for $r = 1, \ldots, R$ **do**
    Simulate $\tilde{\mathbf{x}}^{(r)} \sim p(\cdot \mid \boldsymbol{\theta})$;
  end
  Evaluate $\hat{p}_{\boldsymbol{\theta}}(\mathbf{y})$ using $\{\tilde{\mathbf{x}}^{(r)}\}_{r=1}^{R}$ according to Equation (1.15);
  Set $\hat{p}_{\boldsymbol{\theta}_i}(\mathbf{y}) = \hat{p}_{\boldsymbol{\theta}}(\mathbf{y})$, $\boldsymbol{\theta}_i = \boldsymbol{\theta}$ and $\{\mathbf{x}^{(r)}\}_{r=1}^{R} = \{\tilde{\mathbf{x}}^{(r)}\}_{r=1}^{R}$ with probability

$$\alpha = \min\left\{1, \frac{\hat{p}_{\boldsymbol{\theta}}(\mathbf{y})\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}_{i-1} \mid \boldsymbol{\theta})}{\hat{p}_{\boldsymbol{\theta}_{i-1}}(\mathbf{y})\pi(\boldsymbol{\theta}_{i-1})q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{i-1})}\right\},$$

  otherwise set $\hat{p}_{\boldsymbol{\theta}_i}(\mathbf{y}) = \hat{p}_{\boldsymbol{\theta}_{i-1}}(\mathbf{y})$ and $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1}$.
end

$\hat{p}_{\boldsymbol{\theta}}(\mathbf{y})$ is an unbiased, non-negative estimate of some desired target $p_{\boldsymbol{\theta}}(\mathbf{y})$, then Algorithm 1 is an example of a pseudo-marginal MH algorithm (Andrieu et al., 2009). Note that while the sampling procedure in Algorithm 1 exactly targets the posterior distributions described above, many alternative posterior sampling algorithms exist (see e.g. Beaumont et al., 2009).

We once again emphasise that a major disadvantage to applying the methods described by Grazzini et al. (2017) and Platt (2021) to simulation models in the social sciences – such as ABMs – is that estimating the posterior involves simulating $R \geq 1$ times from the ABM at each proposed parameter value $(\boldsymbol{\theta}^{(i)})_{i=1,\ldots,n}$. Since simulation models in the social sciences are often expensive to simulate, and n is required to be many hundreds of thousands in order to accurately estimate the targeted posterior, such approaches can rapidly lead to a prohibitively large number of required simulations.

1.5.3 Latent variable models

A class of models closely related to approximate Bayesian computation is the class of so-called *latent variable models*. Here, the real data, $\mathbf{y}$, is assumed to be a noisy observation of an unobserved (latent) process $\mathbf{x}$. Thus, in contrast to previous approaches, some discrepancy between $\mathbf{y}$ and $\mathbf{x}$ is to be expected.

Under the most basic scenario, the data obtained from the simulator are identically $\mathbf{x}_t \sim f(\cdot \mid \boldsymbol{\theta})$ for some density function $f(\cdot \mid \boldsymbol{\theta})$ and the total likelihood $p(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} f(\mathbf{x}_t \mid \boldsymbol{\theta})$. The observed data is then assumed to have an error distribution, g_ε, say, such that the contribution made by one observation $\mathbf{y}_t$ to the likelihood function is

$$\tilde{f}(\mathbf{y}_t \mid \boldsymbol{\theta}) = \int g_\varepsilon(\mathbf{y}_t \mid \mathbf{x}_t) f(\mathbf{x}_t \mid \boldsymbol{\theta}) \mathrm{d}\mathbf{x}_t \tag{1.30}$$

and $p(\mathbf{y} \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} \tilde{f}(\mathbf{y}_t \mid \boldsymbol{\theta})$ is the likelihood for the full data set $\mathbf{y}$. We draw the attention of the read to the conceptual similarity to (1.15); however, (1.30) is no longer an approximation but is the exact likelihood under the assumption of measurement error. (Similarly, ABC can be seen as exact if K_ε describes the observational error, see e.g. Wilkinson 2013.) A simple unbiased estimator of (1.30) is

$$\hat{p}_{\boldsymbol{\theta}}(\mathbf{y}) = \prod_{t=1}^{T} \frac{1}{R} \sum_{r=1}^{R} g_\varepsilon\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right), \qquad \mathbf{x}_t^{(r)} \sim f\left(\cdot \mid \boldsymbol{\theta}\right)$$

for some[3] $R \geq 1$, which can then also be used in Algorithm 1, for example, to target the associated approximate posterior. The independence assumption underlying the simulator data is, however, not realistic for ABMs and other time-series models, and incorporating the temporal aspect of data requires further structural assumptions, such as those of hidden Markov models.

1.5.3.1 Hidden Markov Models and particle filters in the social sciences

Particle filters, an instance of a broader class of sequential Monte Carlo (SMC) methods (see Ju et al., 2021, for a recent overview of SMC and their application to epidemiological ABMs) relax the independence assumption imposed on the output of the simulation model and have previously been employed for Bayesian parameter inference for economic ABMs (Lux, 2018, 2021). However, such methods also involve

[3]Note that opposed to earlier cases the choice $R = 1$ is generally not optimal for products of unbiased estimators, see Doucet et al. (2015); Sherlock et al. (2015); Schmon and Gagnon (2021).

making assumptions about the structure of the model – specifically, the assumption of a state space structure with a particular error distribution. Furthermore, previous works have noted the significant computational burden required for SMC methods in the social sciences (Malleson et al., 2020; Lux, 2021), which can arise due to the fact that a number of particles that grows exponentially with the model dimensions is often required to avoid failure modes such as particle collapse.

1.5.4 Modern simulation-based inference methods

In this section, we discuss modern simulation-based inference (SBI) procedures that have emerged more recently than the methods described above, and that are based on estimating the posterior density (Papamakarios and Murray, 2016; Greenberg et al., 2019) or likelihood-to-evidence ratio (Pham et al., 2014; Cranmer et al., 2016; Thomas et al., 2021; Hermans et al., 2020) directly. These approaches have been seen to perform competitive likelihood-free inference with far fewer samples than ABC in well-specified settings (Lueckmann et al., 2021). For these methods, the general approach departs from that of ABC – in which simulation is inherent to the task of constructing a specific posterior distribution $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ – in that the burden of simulation is removed from the task of posterior construction by first learning a density (ratio) *estimator* with powerful function approximators (e.g. neural networks[4]), before using such density (ratio) estimators to cheaply generate posterior samples, without the need for further simulation. Such approaches may be preferable when the simulation budget is constrained, since it has been seen in practice (Lueckmann et al., 2021) that the number of simulations required to obtain accurate density (ratio) estimators is usually far smaller than the number of simulations required to attain a comparably accurate posterior estimate via ABC when the model is well-specified.

1.5.4.1 Density ratio estimation for likelihood-free inference

We first consider density ratio estimation (DRE). It is well-known that probabilistic classification results in density ratio estimates as a by-product (Sugiyama et al., 2012). Within the topic of likelihood-free parameter inference, this fact was to the best of

[4]Note that neural networks are not essential here – the procedures we describe require only flexible function approximators. We frame the discussion around neural networks primarily because they are a convenient choice of flexible function approximators in many settings.

our knowledge first discussed in a Bayesian context by Pham et al. (2014) as a way to estimate the acceptance probability in MH, and in a frequentist context by Cranmer et al. (2016) as a means to performing likelihood ratio tests. Both cases proceed by learning a probabilistic classifier that distinguishes between samples $\mathbf{x} \overset{iid}{\sim} p(\mathbf{x} \mid \boldsymbol{\theta}_0)$ and $\mathbf{x}' \overset{iid}{\sim} p(\mathbf{x}' \mid \boldsymbol{\theta}_1)$, yielding an estimate of the likelihood ratio

$$l(\mathbf{x}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) := \frac{p(\mathbf{x} \mid \boldsymbol{\theta}_0)}{p(\mathbf{x} \mid \boldsymbol{\theta}_1)}. \tag{1.31}$$

Cranmer et al. (2016) propose to use a fixed reference $\boldsymbol{\theta}_1$ and consequently to use the density ratio estimate (1.31) to obtain an maximum likelihood estimate (MLE) for $\mathbf{x}$ by maximising (1.31) with respect to $\boldsymbol{\theta}_0$. Pham et al. (2014) embed this approach into a Bayesian inference framework by using this to estimate the acceptance probability in a MH proposal step as

$$\min \left\{ 1, \frac{\pi(\boldsymbol{\theta}) q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}{\pi(\boldsymbol{\theta}') q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')} l(\mathbf{x}, \boldsymbol{\theta}', \boldsymbol{\theta}) \right\}. \tag{1.32}$$

However, both approaches are sub-optimal for LFI for the following main reasons: it requires retraining a separate classifier at each proposal step in a posterior sampling procedure, which can be expensive; and it requires the generation of sufficiently many samples at each proposal step to accurately train the classifier, which may also be computationally demanding. These drawbacks introduce significant inefficiencies to Bayesian LFI procedures based on this conception of DRE.

DRE as a means to Bayesian LFI was later developed in an early draft of Thomas et al. (2021). The approach proposed here differs from Pham et al. (2014) and Cranmer et al. (2016) in that the authors propose to train a probabilistic classifier at each proposal step, but in this case to distinguish between simulations generated by $p(\mathbf{x} \mid \boldsymbol{\theta})$ and $p(\mathbf{x})$. Such an approach will yield an estimate of the likelihood-to-evidence ratio at that $\boldsymbol{\theta}$. However, it remains the case that a new classifier must be trained at each proposed $\boldsymbol{\theta}$, and sufficiently many training data $\mathbf{x}$ must be generated by that $\boldsymbol{\theta}$ for the classifier to obtain a good density ratio estimate. This approach remains inefficient for these reasons.

Amortised density ratio estimation via binary classification

In the following, we outline *amortised* DRE, as described by Hermans et al. (2020), and discuss how this addresses the two inefficiencies that remain with the approach of Thomas et al. (2021) to Bayesian LFI. In contrast to the approach of Thomas et al. (2021), amortised DRE seeks to obtain a *global* estimator $r : \mathcal{X} \times \boldsymbol{\Theta} \to \mathbb{R}_{\geq 0}$ of the likelihood-to-evidence ratio at $(\mathbf{x}, \boldsymbol{\theta})$:

$$\frac{p\left(\mathbf{x} \mid \boldsymbol{\theta}\right) \pi(\boldsymbol{\theta})}{p\left(\mathbf{x}\right) \pi\left(\boldsymbol{\theta}\right)} = \frac{p\left(\mathbf{x} \mid \boldsymbol{\theta}\right)}{p\left(\mathbf{x}\right)} = \frac{\pi\left(\boldsymbol{\theta} \mid \mathbf{x}\right)}{\pi\left(\boldsymbol{\theta}\right)} =: r(\mathbf{x}, \boldsymbol{\theta}). \tag{1.33}$$

When Bayesian inference is the goal, this then permits evaluation of the posterior density as

$$\pi(\boldsymbol{\theta} \mid \mathbf{x}) = r(\mathbf{x}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta}); \tag{1.34}$$

however, we note that obtaining this density ratio estimate enables alternative inference tasks such as frequentist maximum likelihood estimation and hypothesis testing (Dalmasso et al., 2020).

To obtain a global estimator of the likelihood-to-evidence ratio, a probabilistic binary classifier is trained to distinguish between two sets of training examples:

1. a set of "genuine" examples drawn from the joint distribution, $(\mathbf{x}, \boldsymbol{\theta}) \overset{iid}{\sim} p(\mathbf{x} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta})$, which are then assigned label $z = 1$;
2. a set of "false" examples drawn from the product of the marginals, $(\mathbf{x}, \boldsymbol{\theta}) \overset{iid}{\sim} p(\mathbf{x}) \pi(\boldsymbol{\theta})$, with label $z = 0$.

The difference between the two cases is that the $\mathbf{x}$ are generated by the $\boldsymbol{\theta}$ to which they are paired in the former set of examples, while in the latter set of examples the $\mathbf{x}$ bear no relation to the $\boldsymbol{\theta}$ they are paired with.

The function of a probabilistic classifier trained on such data is to model the probability $d(\mathbf{x}, \boldsymbol{\theta}) := p(z = 1 \mid \mathbf{x}, \boldsymbol{\theta}) \in [0, 1]$; hard classification labels (i.e. decisions regarding the predicted value of z) are obtained when the continuous-valued $d(\mathbf{x}, \boldsymbol{\theta})$ is combined with a decision rule, for example that $z = 1$ should be predicted whenever $d(\mathbf{x}, \boldsymbol{\theta}) > 0.5$. One can show that the *optimal* estimate (Thomas et al., 2021; Hermans et al., 2020) of $d(\mathbf{x}, \boldsymbol{\theta})$ is the value

$$d^*(\mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) + p(\mathbf{x})\pi(\boldsymbol{\theta})}. \tag{1.35}$$

Thus, a good probabilistic classifier trained to distinguish between the two possible types of pairs $(\mathbf{x}, \boldsymbol{\theta})$ will learn a good estimate $\hat{d}(\mathbf{x}, \boldsymbol{\theta})$ of this ratio.

Such a probabilistic classifier will allow us to evaluate the posterior density in this way: by noticing that one can rearrange Equation (1.35) as

$$\pi(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{d^*(\mathbf{x}, \boldsymbol{\theta})}{1 - d^*(\mathbf{x}, \boldsymbol{\theta})}\pi(\boldsymbol{\theta}) := r^*(\mathbf{x}, \boldsymbol{\theta})\pi(\boldsymbol{\theta}), \tag{1.36}$$

where $r^*(\mathbf{x}, \boldsymbol{\theta})$ is the corresponding estimate of the likelihood-to-evidence ratio $p(\mathbf{x} \mid \boldsymbol{\theta})/p(\mathbf{x})$. In practice, of course, only an approximation $\hat{d}(\mathbf{x}, \boldsymbol{\theta})$ will be obtained to the ratio in Equation (1.35), yielding a correspondingly imperfect estimate $\hat{r}(\mathbf{x}, \boldsymbol{\theta})$ of the likelihood-to-evidence ratio. Nonetheless, it is known that training expressive, high-capacity classifiers via the cross-entropy loss

$$\ell\left(\left\{z^{(i)}, \mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}\right\}_{i=1}^{N}\right) = -\sum_{i=1}^{N} \frac{z^{(i)} \log \hat{d}(\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}) + (1 - z^{(i)}) \log\left(1 - \hat{d}(\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)})\right)}{N},$$

can yield classifiers with good probability estimates, and thus good estimates of $r^*(\mathbf{x}, \boldsymbol{\theta})$.

A generalisation to multi-class classification

In more recent work, Durkan et al. (2020) demonstrate that the above approach to density ratio estimation can be generalised to the problem of training a probabilistic classifier to identify the *correct* $(\mathbf{x}, \boldsymbol{\theta}^{(i)})$ pair from a batch $\{(\mathbf{x}, \boldsymbol{\theta}^{(b)})\}_{b=1}^{B}$ of B pairs that otherwise contain only "incorrect" pairs, where "correct" and "incorrect", respectively, correspond to having been drawn from the joint distribution $p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ and from the product of the marginal distributions $p(\mathbf{x})\pi(\boldsymbol{\theta})$. In this case, the optimal estimate of the correct class probability (assuming that each class is equally likely *a*

priori) is

$$p\left(c = i \mid \mathbf{x}, \{\boldsymbol{\theta}^{(b)}\}_{b=1}^{B}\right) = \frac{\pi(\boldsymbol{\theta}^{(i)} \mid \mathbf{x}) \prod_{b \neq i} \pi(\boldsymbol{\theta}^{(b)})}{\sum_{b=1}^{B} \pi(\boldsymbol{\theta}^{(b)} \mid \mathbf{x}) \prod_{b' \neq b} \pi(\boldsymbol{\theta}^{(b')})} \tag{1.37}$$

$$= \frac{\pi(\boldsymbol{\theta}^{(i)} \mid \mathbf{x})/\pi(\boldsymbol{\theta}^{(i)})}{\sum_{b=1}^{B} \pi(\boldsymbol{\theta}^{(b)} \mid \mathbf{x})/\pi(\boldsymbol{\theta}^{(b)})}. \tag{1.38}$$

Thus, by comparison with Equation (1.38), training a neural network f_ϕ on the loss function

$$\ell(\phi) = -\log \frac{\exp f_\phi(\mathbf{x}, \boldsymbol{\theta}^{(i)})}{\sum_{b=1}^{B} \exp f_\phi(\mathbf{x}, \boldsymbol{\theta}^{(b)})} \tag{1.39}$$

for each $(\mathbf{x}, \boldsymbol{\theta}) \sim p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ will induce $f_\phi(\mathbf{x}, \boldsymbol{\theta})$ to learn the value $f_\phi(\mathbf{x}, \boldsymbol{\theta}) = \log(\pi(\boldsymbol{\theta} \mid \mathbf{x})/\pi(\boldsymbol{\theta}))$, thus recovering an estimate of the desired density ratio. This can be extended to a batch of R "correct" data-parameter pairs as

$$\mathcal{L}(\phi) = -\frac{1}{R} \sum_{r=1}^{R} \log \frac{\exp f_\phi(\mathbf{x}^{(r)}, \boldsymbol{\theta}^{(r)})}{\sum_{b_r=1}^{B_r} \exp f_\phi(\mathbf{x}^{(r)}, \boldsymbol{\theta}^{(b_r)})}, \tag{1.40}$$

where the terms in the denominator are labelled with b_r to account for the possibility that the contrasting ("incorrect") set of parameters may be different for different "correct" pairs $(\mathbf{x}^{(r)}, \boldsymbol{\theta}^{(r)})$. The authors further demonstrate that this can yield more accurate density ratio estimators in some cases, although such an approach may be more computationally expensive (since each $\mathbf{x}$ is now passed through the network $B > 2$ times, once for each $\boldsymbol{\theta}$ in the batch).

1.5.4.2 Neural posterior estimation

In this section, we provide an introduction to neural conditional density estimators and their use in posterior estimation tasks, forming a class of methods known as neural posterior estimation (NPE). The core idea underlying this class of simulation-based Bayesian inference techniques is to use such conditional density estimators to model the parameter posterior density directly. While various conditional density estimators can be used for this purpose, e.g. mixture density networks (Bishop, 1994; Papamakarios and Murray, 2016), we focus here on the case of **normalising flows** (Tabak and Vanden-Eijnden, 2010; Tabak and Turner, 2013; Rezende and Mohamed, 2015) due to their widespread use in various density estimation tasks, including in

the areas of image generation (e.g. Kingma and Dhariwal, 2018) and physics (e.g. Noé et al., 2019).

Normalising flows

Normalising flows involve transforming a simple base distribution into a more complicated one, often with the use of neural networks. Consider a random variable $\mathbf{U} \sim p_{\mathbf{U}}$, where $p_{\mathbf{U}}$ is a probability distribution chosen to be a "simple" base distribution, in the sense that it is easy to both generate samples from $p_{\mathbf{U}}$ and to evaluate $p_{\mathbf{U}}(\mathbf{u})$ for any $\mathbf{u}$. Now consider the transformed random variable $\mathbf{X} = g(\mathbf{U})$, where g is a differentiable, invertible function with differentiable inverse $f := g^{-1}$. Then, by the change of variables formula, we have

$$p_{\mathbf{X}}(\mathbf{x}) = p_{\mathbf{U}}\left(f(\mathbf{x})\right) \left| \det J_f(\mathbf{x}) \right| \tag{1.41}$$

where J_f is the Jacobian of f. Sampling from $p_{\mathbf{X}}$ then simply involves sampling a value $\mathbf{u}$ from the base distribution $p_{\mathbf{U}}$ and immediately obtaining $\mathbf{x} = g(\mathbf{u})$. Evaluating $p_{\mathbf{X}}(\mathbf{x})$ also then simply involves evaluation of the right-hand side of Equation (1.41). The above may be extended easily to a composition, or *flow*, $g = g_n \circ g_{n-1} \circ \cdots \circ g_1$ of differentiable and invertible functions g_i with inverses f_i, for which we now have that

$$p_{\mathbf{X}}(\mathbf{x}) = p_{\mathbf{U}}\left(f_1\left(\ldots f_{n-1}\left(f_n(\mathbf{x})\right)\ldots\right)\right) \left| \prod_{i=1}^{n} \det J_{f_i}(\mathbf{x}) \right|. \tag{1.42}$$

Given samples from $p_{\mathbf{X}}$, the question of how to choose a base distribution $p_{\mathbf{U}}$ and a series of functions $g_1, \ldots, g_n$ such that the distribution of the samples is modelled accurately arises. The idea behind normalising flows is to *learn* this sequence of transformations, that is, to have each f_i be a flexible transformation f_{ϕ_i} with trainable parameters ϕ_i. The resulting density estimator q_ϕ can be written

$$q_\phi(\mathbf{x}) = p_{\mathbf{U}}\left(f_{\phi_1}\left(\ldots f_{\phi_{n-1}}\left(f_{\phi_n}(\mathbf{x})\right)\ldots\right)\right) \left| \prod_{i=1}^{n} \det J_{f_{\phi_i}}(\mathbf{x}) \right|, \quad \text{where} \quad \phi = (\phi_n, \ldots, \phi_1). \tag{1.43}$$

The simple base distribution is then often taken to be a standard normal distribution[5] of the same dimension as the target distribution, and the flexible transformations f_{ϕ_i} are typically neural networks with a structure designed to guarantee the required

[5]For this reason, the composition $f = f_1 \circ \cdots \circ f_{n-1} \circ f_n$ is often referred to as the *normalising* flow, since it is a flow of transformations resulting (typically) in a Gaussian base distribution.

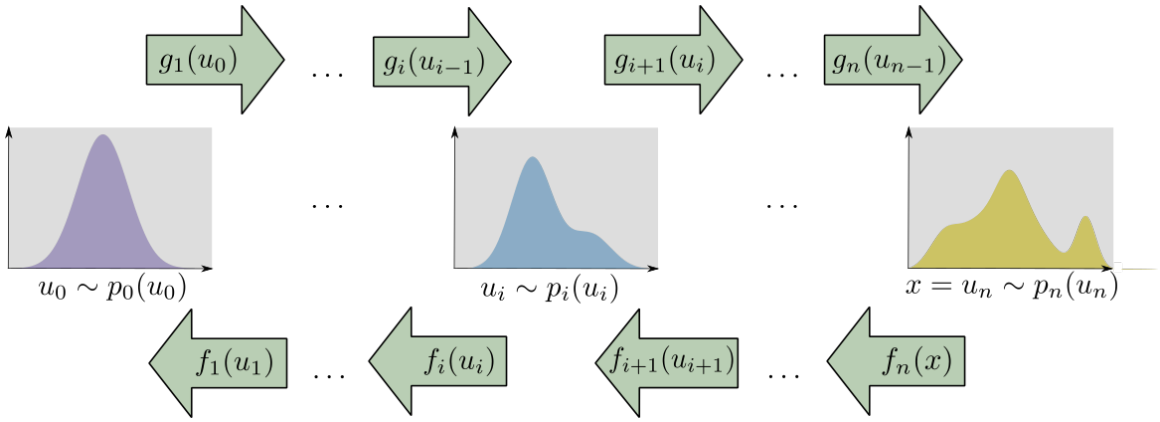


Figure 1.1: Schematic of a normalising flow. Sampling and density evaluation are performed via the processes illustrated at the top and bottom of the figure, respectively.

invertibility and differentiability. The trainable parameters $\phi = \{\phi_i : 1 \leq i \leq n\}$ are optimised by maximising the log-likelihood of the data $\mathbf{x}^{(r)} \overset{iid}{\sim} p_{\mathbf{X}}, r = 1, \ldots, R$,

$$\hat{\phi} = \arg\max_{\phi} \sum_{r=1}^{R} \log q_\phi \left(\mathbf{x}^{(r)}\right), \tag{1.44}$$

which is equivalent to minimising the finite-sample estimate of the KL divergence between q_ϕ and the true density. Normalising flows are typically implemented in machine learning libraries that support automatic differentiation such as `pytorch` (Paszke et al., 2019) or `TensorFlow` (Abadi et al., 2016) allowing the effective use of gradient-based optimisation techniques, such as Adam (Kingma and Ba, 2014). We provide a schematic of the sampling (flow, right-pointing arrows) and density evaluation (normalising flow, left-pointing arrows) processes in Figure 1.1, in which the simple, left-most distribution is morphed over successive steps into the more complex, right-most distribution.

Normalising flows for neural posterior estimation

Normalising flows may also be extended to the case of *conditional* density estimation (see e.g. Papamakarios and Murray, 2016; Lueckmann et al., 2017; Papamakarios et al., 2019; Greenberg et al., 2019). In this way, we can use a normalising flow to estimate the posterior density $\pi(\boldsymbol{\theta} \mid \mathbf{x})$ associated with a simulation model by following the same procedure as above and finding the neural network parameters as

$$\hat{\phi} = \arg\max_{\phi \in \Phi} \sum_{r=1}^{R} \log q_\phi \left(\boldsymbol{\theta}^{(r)} \mid \mathbf{x}^{(r)}\right), \tag{1.45}$$

where $(\mathbf{x}^{(r)}, \boldsymbol{\theta}^{(r)}) \overset{iid}{\sim} p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}), r = 1, \ldots, R$. Once again, this is equivalent to minimising the finite-sample estimate of the KL divergence between the true joint density $p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ and the approximated joint density, $q_\phi(\boldsymbol{\theta} \mid \mathbf{x})p(\mathbf{x})$. Importantly, this framework allows us to learn a *single* conditional density estimator for the posterior density function across data $\mathbf{x}$, rather than having to undergo the expensive procedure of training a separate density estimator for each point evaluation of the posterior density as in e.g. Platt (2021).

1.5.4.3 Sampling with the neural posterior and density ratio estimators

After successful training, the posterior density estimator $q_{\hat{\phi}}(\boldsymbol{\theta} \mid \mathbf{x})$ obtained from NPE is a parametric approximation of the true posterior distribution, which can then be used to generate *iid* samples $\boldsymbol{\theta} \sim q_{\hat{\phi}}(\boldsymbol{\theta} \mid \mathbf{x})$ or to evaluate the posterior density on a pointwise basis in the ways described in Section 1.5.4.2. While ratio estimation similarly permits a pointwise evaluation of the posterior distribution as $\pi(\boldsymbol{\theta}) \exp\left(f_{\hat{\phi}}(\mathbf{x}, \boldsymbol{\theta})\right)$, it requires a further inference step using, for example, Metropolis–Hastings to generate samples from the same posterior. However, the upfront training of the ratio estimator eliminates the need for expensive simulation from the ABM, reducing the run-time significantly in comparison to alternative approaches such as those described above.

1.5.4.4 Round-based training of neural density ratio and neural posterior estimation algorithms

The procedures described in Sections 1.5.4.1 and 1.5.4.2 are framed as single density (ratio) estimation tasks. This means that a single dataset of R data-parameter pairs $(\mathbf{x}^{(r)}, \boldsymbol{\theta}^{(r)}), r = 1, \ldots, R$ is created in the following way:

$$\begin{aligned} \boldsymbol{\theta}^{(r)} &\sim \pi(\boldsymbol{\theta}), && \text{sample from the prior;} \\ \mathbf{x}^{(r)} &\sim p(\mathbf{x} \mid \boldsymbol{\theta}^{(r)}), && \text{simulate using the draws from the prior.} \end{aligned}$$

Subsequently, DRE and NPE algorithms are trained on the *whole dataset* to find either the likelihood-to-evidence ratio or the posterior directly following, respectively, (1.40) or (1.45). Density estimators trained in this way are referred to as *amortised* density

estimators, which reflects the fact that they may be used to construct posterior distributions for any data $\mathbf{x}$ without further simulation from the simulator or the need to train multiple density estimators.

The disadvantage of this one-stage approach is that all training simulations from the simulator are generated by parameters drawn from the prior distribution. In many cases, however, interest lies only in the posterior for the *particular observation* $\mathbf{y}$, which can be concentrated on specific subregions of the entire space covered by the prior density. It can then be preferable to narrow down the search space of good candidate values for the parameter $\boldsymbol{\theta}$ to subregions of the parameter space that could most plausibly have generated $\mathbf{y}$. By doing so, the density (ratio) estimator will be presented with a less varied range of dynamics between which it must learn to distinguish, allowing it to develop a more refined approximation of the density (ratio) in regions of high posterior density. This can facilitate more rapid learning and potentially reduce the number of training examples that must be simulated by the simulator, which may be useful for expensive simulation models in the social sciences such as ABMs.

Significant effort has thus been extended towards the constructions of "round"-based approaches to training density (ratio) estimators for SBI (see e.g. Papamakarios et al., 2019; Greenberg et al., 2019; Hermans et al., 2020; Durkan et al., 2020). The idea is to split the total budget of R simulations into subsets of size $N_1, N_2, \ldots$ such that $\sum_i N_i = R$. In the first step, N_1 data points are created as before and a posterior approximation $q_{\hat{\phi}}(\boldsymbol{\theta} \mid \mathbf{x})$ is constructed. Subsequently, in round $i \geq 2$, we create new data $(\mathbf{x}^{(r)}, \boldsymbol{\theta}^{(r)}), r = 1, \ldots, N_i$ using

$$\begin{aligned} \boldsymbol{\theta}^{(r)} &\sim q_{\hat{\phi}}^{(i-1)}(\boldsymbol{\theta} \mid \mathbf{x}), && \text{sample from round-}(i-1)\text{ posterior;} \\ \mathbf{x}^{(r)} &\sim p(\mathbf{x} \mid \boldsymbol{\theta}^{(r)}), && \text{simulate using the draws from the round-}(i-1)\text{ posterior.} \end{aligned}$$

The posterior $q_{\phi}(\boldsymbol{\theta} \mid \mathbf{x})$ may now be retrained using (a combination of the old and) the new samples, and the process can be repeated for as many rounds as necessary. An identical process can be followed for density ratio estimation, with the exception that parameters in round i are drawn using the likelihood-to-evidence ratio obtained from round $i-1$ and, for example, Metropolis–Hastings.

While the round-based approach is appealing, it does not come without additional challenges. In particular, the joint distribution of the example pairs $(\mathbf{x}^{(r)}, \boldsymbol{\theta}^{(r)})$ is no longer $p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ for data sampled after the first round. In practice, this needs

to be accounted for during training by the use of additional weighing factors, such as those appearing in Equation (1.46) (cf. Equation (1.45)). We refer the interested reader to the following papers for further details on this matter: Papamakarios and Murray (2016); Lueckmann et al. (2017); Greenberg et al. (2019).

When training of NPE and DRE proceeds in this round-based fashion, these methods are often referred to as sequential neural posterior estimation (SNPE) and sequential neural ratio estimation (SNRE), respectively, where the prefix "sequential" indicates that a round-based training design of the density (ratio) estimators has taken place and that the resulting density (ratio) estimators are no longer amortised.

1.5.4.5 Training procedures for neural density (ratio) estimators

To summarise and conclude this section, we follow Durkan et al. (2020) and Greenberg et al. (2019) and provide in Algorithms 2 and 3 the procedure for training neural density ratio and neural posterior estimators for SBI over multiple rounds.

1.5.5 Summary statistics in simulation-based inference

For many LFI methods, it can be necessary to reduce high-dimensional data $\mathbf{x}$ into summary statistics $\mathbf{s}(\mathbf{x})$. A number of approaches for doing so have been explored in both the ABC and more modern SBI literature. We discuss some common examples below.

1.5.5.1 Best subset selection

A popular approach in ABC, and proposed in Joyce and Marjoram (2008), is to construct, through some means or other, a large set of p candidate statistics $\mathbf{s} = (s_1, \ldots, s_p)$ from which a subset are to be retained according to a criterion. The criterion considered by Joyce and Marjoram (2008) is motivated by the notion of sufficiency, and by recalling that the definition of classical sufficiency in Definition 2 is that the conditional probability (density) of some data is conditionally independent of the parameters $\boldsymbol{\theta}$ when also conditioned on a sufficient statistic. With this in mind, the authors consider a scheme in which an approximately sufficient subset of the

Algorithm 2: Training SNRE (see Durkan et al. (2020, Algorithm 1)).

Input: prior $\pi(\boldsymbol{\theta})$, simulator $p(\mathbf{x} \mid \boldsymbol{\theta})$, observation $\mathbf{y}$, density (ratio) estimator f_ϕ, number of rounds M, number of simulations per round N, minibatch size B, contrasting set size K;
Result: Trained density (ratio) estimator
Set $\tilde{\pi}_0(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$, dataset $\mathcal{D} = \{\}$;
for $m = 0, \ldots, M-1$ **do**
  Sample $\boldsymbol{\theta}_n \overset{iid}{\sim} \tilde{\pi}_m(\boldsymbol{\theta})$, $n = 1, \ldots, N$;
  Simulate $\mathbf{x}_n \sim p(\mathbf{x} \mid \boldsymbol{\theta}_n)$, $n = 1, \ldots, N$;
  Append the dataset:

$$\mathcal{D} := \mathcal{D} \cup \bigcup_{n=1}^{N} \{(\mathbf{x}_n, \boldsymbol{\theta}_n)\};$$

  until *convergence* **do**
    Sample minibatch $\{(\mathbf{x}_b, \boldsymbol{\theta}_b)\}_{b=1}^{B}$ uniformly from $\mathcal{D}$;
    For each minibatch pair, draw $0 < K < B$ parameters $\tilde{\boldsymbol{\theta}}_k$ from elsewhere in the training data $\mathcal{D}$, $k = 1, \ldots, K$;
    Evaluate the finite-sample multinomial logistic loss

$$\mathcal{L}(\phi) = -\frac{1}{B} \sum_{b=1}^{B} \log \frac{\exp\left(f_\phi\left(\mathbf{x}_b, \boldsymbol{\theta}_b\right)\right)}{\exp\left(f_\phi\left(\mathbf{x}_b, \boldsymbol{\theta}_b\right)\right) + \sum_{k=1}^{K} \exp\left(f_\phi\left(\mathbf{x}_b, \tilde{\boldsymbol{\theta}}_k\right)\right)};$$

    Update trainable parameters ϕ on the basis of $\mathcal{L}(\phi)$
  end
  Set $\tilde{\pi}_{m+1}(\boldsymbol{\theta}) \propto \exp\left(f_\phi\left(\mathbf{y}, \boldsymbol{\theta}\right)\right)$.
end

initial p statistics is arrived at by adding an element of $\mathbf{s}$ one by one, and assigning the following "score" to the kth element that is considered from $\mathbf{s}$:

$$\delta_{\sigma_k} = \sup_{\boldsymbol{\theta}} \log p(s_{\sigma_k} \mid s_{\sigma_1}, \ldots, s_{\sigma_{k-1}}, \boldsymbol{\theta}) - \inf_{\boldsymbol{\theta}} \log p(s_{\sigma_k} \mid s_{\sigma_1}, \ldots, s_{\sigma_{k-1}}, \boldsymbol{\theta}), \tag{1.47}$$

where σ is some permutation of the indices $\{1, \ldots, p\}$ and σ_i is the ith element of the permutation. Then, if the $s_{\sigma_1}, \ldots, s_{\sigma_{k-1}}$ are "ϵ-sufficient" for s_{σ_k} – meaning that $\delta_{\sigma_k} \leq \epsilon$ for a user-specified value of ϵ – then s_{σ_k} is *not* included, since it does not contribute enough new information to the set $s_{\sigma_1}, \ldots, s_{\sigma_{k-1}}$. Otherwise, s_{σ_k} is deemed to contain enough additional information to include in the subset of statistics that are retained. The authors provide details on a practical approach to performing this procedure, which cannot be applied as-is since the values $\log p(s_{\sigma_k} \mid s_{\sigma_1}, \ldots, s_{\sigma_{k-1}}, \boldsymbol{\theta})$ will in general not be known due to the intractability of the likelihood function.

Algorithm 3: Training SNPE (see Greenberg et al., 2019, Algorithm 1).

Input: prior distribution $\pi(\boldsymbol{\theta})$, simulator $p(\mathbf{x} \mid \boldsymbol{\theta})$, observation $\mathbf{y}$, conditional density estimator q_ϕ, number of rounds M, number of simulations per round N;
Result: Trained conditional density estimator
Set $\tilde{\pi}_0(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$, dataset $\mathcal{D} = \{\}$;
for $m = 0, \ldots, M-1$ **do**
  Sample $\boldsymbol{\theta}^{(n)} \overset{iid}{\sim} \tilde{\pi}_m(\boldsymbol{\theta})$, $n = 1, \ldots, N$;
  Simulate $\mathbf{x}^{(n)} \sim p\left(\mathbf{x} \mid \boldsymbol{\theta}^{(n)}\right)$, $n = 1, \ldots, N$;
  Append the dataset:

$$\mathcal{D} := \mathcal{D} \cup \bigcup_{n=1}^{N} \left\{ \left(\mathbf{x}^{(n)}, \boldsymbol{\theta}^{(n)}\right) \right\};$$

  until *convergence* **do**
    Evaluate the loss function

$$\mathcal{L}(\phi) = -\sum_{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{D}} \log\left(q_\phi(\boldsymbol{\theta} \mid \mathbf{x}) \frac{\tilde{\pi}_m(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \frac{1}{Z(\mathbf{x}, \phi)} \right) \tag{1.46}$$

    where $Z(\mathbf{x}, \phi)$ is a normalising factor;
    Update trainable parameters ϕ on the basis of $\mathcal{L}(\phi)$
  end
  Set $\tilde{\pi}_{m+1}(\boldsymbol{\theta}) := q_\phi(\mathbf{y}, \boldsymbol{\theta})$.
end

There are a number of limitations to this approach, however. Most obvious is the sensitivity of this approach to the choice of ϵ in the definition of "ϵ-sufficiency". Moreover, it assumes that it is easy to write down an initial set of p summary statistics that are themselves approximately sufficient and are therefore informative of the parameter to be inferred. This therefore does not straightforwardly get around the more fundamental problem of identifying suitable and informative summary statistics for any data, and in particular for time-series data.

1.5.5.2 Projection methods

A number of methods exist that may appropriately be grouped under the term "projection" methods, which is a term that has been used in previous reviews of summary statistic contruction for LFI (see e.g. Blum et al., 2013). While the objective of such

approaches are also to express the original dataset in a form that is more useful for LFI, these methods differ from the "best subset selection" method described above in the sense that the original data is in general transformed through a (sequence of) map(s), rather than having individual summary statistics within an original collection of summary statistics retained for later use. We summarise below some of the main approaches within this class of methods.

Semi-automatic summary statistic construction Mentioned briefly in Section 1.5.1 is the semi-automatic approach to summary statistic construction in LFI proposed in Fearnhead and Prangle (2012). The authors argue that optimality of a set of summary statistics may be considered with respect to the goal of minimising the expected posterior loss in ABC. By considering a quadratic loss function between point estimates generated by the ABC posterior and the ground-truth parameter, the summary statistics $\mathbf{s}(\mathbf{y}) = \mathbb{E}[\boldsymbol{\theta} \mid \mathbf{y}]$ are shown to be optimal with respect to this loss. While the posterior is not known and must be estimated through ABC or some other scheme, the posterior *mean* may be estimated for the simulation model by performing a vector-valued regression of $\boldsymbol{\theta}^{(i)}$ onto $g\left(\mathbf{x}^{(i)}\right)$ for training data $\left\{\left(\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}\right)\right\}_{i=1}^{Q} \sim p(\mathbf{x}, \boldsymbol{\theta})$ and an initial set of candidate summary statistics $g(\cdot)$.

This has been an influential approach within ABC, and has been used widely such as in Wong et al. (2018), where the authors perform the regression task using a multilayer perceptron, and in Åkesson et al. (2020), where the authors perform the semi-automatic projection method for time-series data using a convolutional neural network architecture. A third example that is relevant to time-series data is the partially exchangeable networks (PENs) architecture (Wiqvist et al., 2019), in which a neural network architecture is proposed that exploits certain model symmetries and is constructed as follows. Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n), \mathbf{x}_i \in \mathcal{X}$ be sequential data generated by a stochastic process of Markov order r, and A be a metric space. A partially exchangeable network model $F : \mathcal{X}^n \to A$ consist of two networks $\phi : \mathcal{X}^{r+1} \to \mathbb{R}$ and $\rho : \mathcal{X}^r \times \mathbb{R} \to A$ combined as

$$F(\mathbf{x}) = \rho\left(\mathbf{x}_{1:r}, \sum_{i=1}^{n-r} \phi\left(\mathbf{x}_{i:(i+r)}\right)\right).$$

The network F is then trained in the semi-automatic manner to learn the posterior mean as a summary statistic. There are limited additional examples outside of ABC:

for example, Dinev and Gutmann (2018) use a convolutional neural network to learn $\mathbf{s}(\mathbf{x}) = \mathbb{E}[\boldsymbol{\theta} \mid \mathbf{x}]$, which are then used as predictors in a logistic regression model for DRE.

A considerable drawback of this approach, however, is that the resultant summary statistics are not in general sufficient, and may be far from sufficient given the restriction of the method to finding the posterior mean as summary statistic, rather than permitting the regression function to find summary statistics of higher dimensionality than $\dim(\boldsymbol{\theta})$ for greater expressivity. Further, in the original approach described in Fearnhead and Prangle (2012), the experimenter must – similarly to the case of "best subset methods" described above – identify an original set of summary statistics that are informative of the parameters in order for the learned regression function to be accurate. This remains difficult, in particular for high-dimensional and highly dependent data such as time-series. This may be somewhat mitigated by the use of neural networks, which can be designed to operate on the full dataset directly; a trade-off exists, however, since in such cases sufficient training data must be simulated and provided to the network in order to accurately perform the regression task. As we will discuss in later chapters, this can be a prohibitive assumption for certain simulation models that will be of interest both in the social sciences and more broadly, since such models can be particularly expensive to run.

Approximately sufficient statistics with mutual information maximisation More recently, Chen et al. (2020) explored the possibility of learning approximately sufficient, mutual information-maximising summary statistics with neural networks. The proposed method is based on the fact that sufficient statistics $s : \mathcal{X} \to \mathcal{S} \subseteq \mathbb{R}^d$ for a model with likelihood function $p(\mathbf{x} \mid \boldsymbol{\theta})$ producing data in some space $\mathcal{X}$ is one that satisfies

$$s = \arg \max_{s' : \mathcal{X} \to \mathcal{S}} I(\boldsymbol{\theta}; s'(\mathbf{x})), \tag{1.48}$$

where

$$I(x; y) = \mathrm{KL}\left(p(x, y) || p(x)p(y)\right) \tag{1.49}$$

is the mutual information between two random variables x and y (Proposition 1 Chen et al., 2020). Consequently, the authors propose to use the mutual information – or, alternatively, a more stable surrogate for the mutual information such as the Jensen-Shannon divergence – between $\boldsymbol{\theta}$ and a learned summary of $\mathbf{x}$, where $(\boldsymbol{\theta}, \mathbf{x}) \sim p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$, as the objective function in a summary statistic learning procedure. The

authors note that the dimensionality d of the learned summary statistic can be as large as the experimenter desires, and demonstrate that taking $d = 2 \cdot \dim(\boldsymbol{\theta})$ can produce competitive performance on an Ornstein-Uhlenbeck process and other non-time-series models when this approach is incorporated into both ABC methods and more modern neural SBI methods.

This approach is appealing due to its basis in the notion of sufficiency, and due to the increased flexibility compared to e.g. semi-automatic summary statistic learning, resulting from the fact that it enables us to learn a summary statistic vector of dimensionality greater than $\dim(\boldsymbol{\theta})$ if this is desired. However, this method shares a drawback present in the case of the semi-automatic approach to summary statistic learning in the context of simulation modelling in the social sciences: it will require sufficient training data from the simulator in order to accurately learn suitable approximately sufficient statistics. For some simulation models in the social sciences – such as expensive agent-based models – the required simulation budget may be prohibitively large, limiting the applicability of this method. The authors also consider only a fully-connected neural network architecture and note that future work is needed to address the problem of designing a problem-specific architecture for the summary statistic-learning network. Considering appropriate architectures for more complex data, such as time-series data, is therefore a useful contribution to the field of summary statistic-learning for SBI.

Embedding networks in neural simulation-based inference The methods described above remain applicable to NPE and DRE. NPE and DRE are however particularly interesting in that they permit the concurrent learning of both summary statistics and densities/ratios by augmenting the density (ratio) estimator with an initial *embedding network* (Lueckmann et al., 2017). More precisely, suppose we have a neural density (ratio) estimator q_ϕ with trainable parameters ϕ, which consumes a data-parameter pair to produce an estimate of the (possibly unnormalised) posterior density at that pair. An *embedding* neural network produces a learnable summary statistic $\mathbf{s}_\varphi$ which prefixes the density (ratio) estimator and performs some initial operations on the raw data $\mathbf{x}$ before passing it through the density (ratio) estimator. The (possibly unnormalised) posterior density is then estimated as $q_\phi\left(\boldsymbol{\theta} \mid \mathbf{s}_\varphi(\mathbf{x})\right)$. The parameters ϕ and φ of this composite summary-learning/posterior-estimating network can then be trained in an end-to-end fashion on the same loss function, which has been seen to produce competitive results (Greenberg et al., 2019).

Table 1.1: Summary of network sizes and simulation budgets for summary statistic learning in previous works.

Authors	Learning method	Net size	Budget
Jiang et al. (2017)	Posterior mean	$\sim 3 \times 10^4$	10^6
Lueckmann et al. (2017)	Embedding network	$\sim 2 \times 10^3$	$\sim 10^4$
Dinev and Gutmann (2018)	Posterior mean	8,422	10^5
Greenberg et al. (2019)	Embedding network	$\sim 3 \times 10^4$	$10^3 - 10^4$
Chen et al. (2020)	Mutual information	$\sim 1.5 \times 10^4$	$10^3 - 10^4$

Such an approach is appealing since it obviates the need to establish two learning tasks – the task of learning summary statistics and the task of learning a good estimate of the posterior density – and subsumes the task of learning good summary statistics into the task of learning a good estimate of the posterior density. In this way, NPE and neural ratio estimation (NRE) offer a convenient approach to incorporating inductive biases through the use of an appropriate neural architecture and to learning summary statistics in an end-to-end fashion without the additional complications associated with alternative SBI techniques. However, as we will again see in later chapters, learning relevant features/summary statistics from neural networks in this way in scenarios where simulation budgets are prohibitively low can remain challenging.

1.5.5.3 Summary of summary statistic learning methods

For later reference, we tabulate some of the key works involving learning summary statistics for time-series data in LFI settings. We list for each the adopted training scheme, the number of trainable parameters in each case (each involved neural networks), and the assumed simulation budgets in Table 1.1. Learning method "posterior mean" refers to the summary statistic-learning method discussed in Fearnhead and Prangle (2012), while learning methods "embedding network" and "mutual information" refer to learning summary statistics by training the summary-learning network on the same loss as the neural density (ratio) estimator (see e.g. Lueckmann et al., 2017) and on some surrogate for the mutual information between the resultant summary statistics and $\boldsymbol{\theta}$ (Chen et al., 2020).

1.5.6 Key challenges

Based on the above review of the literature we identify two common themes that form key challenges in the literature on simulation-based inference:

1. the challenge of dealing appropriately with time-series data of different kinds. In both classical ABC and the more modern approaches to SBI, high-dimensional data such as potentially multivariate time-series data must typically be expressed as a comparatively low-dimensional vector of summary statistics, but there is a significant risk of incurring a large loss of information that is relevant to the parameter inference task at hand. While approaches exist, the question of how best to perform this dimensionality-reduction step for time-series data is unclear and remains a challenge to SBI. Separately, in ABC, there is an opportunity to devise distances between time-series data of different kinds, since the literature on ABC primarily focuses on the case of *iid* observations, but it is unclear how this may be done using existing approaches. Properly handling time-series data of different kinds in SBI is therefore one key challenge that we identify;

2. the challenge of reducing the simulation burden associated with different SBI methods for time-series simulators. Significant progress has been seen in this direction with the introduction of modern SBI methods, in comparison to ABC: the former can produce more accurate inferences than ABC with far fewer simulations[6], but it remains the case that certain simulation models – for example, large ABMs in the social sciences – will benefit or require SBI methods that are even more simulation efficient. Reducing the simulation burden further for time-series simulators is therefore a second key challenge we identify, since this will more readily enable uptake of these methods within the computational social science community, where models often generate time-series data and can be expensive to simulate.

[6] In the well-specified setting, at least; see Part IV for a discussion.

1.6 Thesis aims, outline, and notation

1.6.1 Thesis aims

As described previously, we are concerned in this thesis with methods for performing likelihood-free Bayesian inference. More specifically, we are concerned in this thesis with developing likelihood-free Bayesian inference procedures for approximating Equation (1.2) when the simulation model generates sequential data: data that has a natural ordering, usually because the simulation models the dynamics of a system evolving over time. This is a common scenario encountered generally across application domains – including in economics and the social sciences, which is our main area of interest – and such simulation models and inference procedures deserve special attention due to the fact that data points generated by these simulators cannot be treated as independent and identically distributed (*iid*), which complicates the analysis.

For this purpose, an object arising in stochastic analysis and the theory of controlled differential equations known as the *path signature* features heavily in this thesis as a principled means to automatically extract summary statistics and evoke a notion of distance between time-series $\mathbf{x}$ generated by simulation models and observed time-series $\mathbf{y}$. We will discuss and present novel experiments showing how the path signature can be used in two well-known procedures for likelihood-free Bayesian inference: approximate Bayesian computation, in which – broadly speaking – high (resp. low) parameter posterior density is assigned to regions of the parameter space that tend to generate simulations that closely (resp. poorly) match the observed data, according to a chosen notion of distance; and density ratio estimation, in which the intractable likelihood-to-evidence ratio appearing in Equation 1.2 is approximated using the class probability estimates obtained from a probabilistic classifier.

Finally, we will motivate and present experiments on the use of recently developed neural density (ratio) estimation methods for approximating simulation models' parameter posterior distributions in economics and the social sciences, such as agent-based models. As we will discuss, the literature on parameter estimation methods for agent-based models in economics and the social sciences is relatively under-developed, and the most widely employed methods are in certain important ways poorly suited to such simulators. Specifically, this is due to the fact that agent-based economic

simulation models are typically expensive dynamical models generating complex and high-dimensional time-series as output, whereas the most widely known Bayesian parameter estimation methods: (a) typically require many hundreds of thousands of calls to the simulator to estimate any single posterior $\pi(\boldsymbol{\theta} \mid \mathbf{y})$; (b) seek – either implicitly or explicitly – to model the complex simulation output, rather than the arguably simpler task of modelling the posterior density directly; and/or (c) are unsuitable for time-series data of different kinds without further adjustment.

In summary, the work presented in this thesis is aimed towards developing likelihood-free Bayesian parameter inference techniques that satisfy one or more of the following conditions:

1. the method is suitable for univariate time-series data;
2. the method is suitable for more complex time-series, for example: vector-valued, or graph-valued time-series consisting of regularly or irregularly spaced observations;
3. the method is suitable for expensive simulation models.

A final aim is to help to bridge the gap between the simulation-based inference community in computational statistics and machine learning with the agent-based modelling community in economics and the social sciences, in order to ensure the uptake of state-of-the-art inference methods within the latter community.

1.6.2 Thesis outline

This thesis is structured as follows. In Part II, we will present and discuss our work on the use of path signatures in LFI procedures. This will consist of an introductory chapter on path signatures in Chapter 2, while Chapters 3 through to 5 will focus on their novel use in approximate Bayesian computation and LFI by density ratio estimation, respectively. In each Chapter, existing approaches will be reviewed. In Part III, we will present investigations into the application and efficacy of neural density (ratio) estimation methods for estimating simulation models in economics and the social sciences, such as agent-based models, and demonstrate that this new generation of methods offers significant advantages compared to more classical LFI

techniques in the setting of ABMs. Finally, we will conclude and discuss future avenues of work in Part IV.

1.6.3 Notation

Throughout this thesis, we will use π to denote probability density functions over continuous simulator parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^n$, $n \in \mathbb{N}$, for example as the prior density $\pi(\boldsymbol{\theta})$. Correspondingly, we will use p to denote a probability density over data denoted $\mathbf{x}, \mathbf{y}$, or occasionally $\mathbf{z}$, e.g. as in the likelihood function $p(\mathbf{x} \mid \boldsymbol{\theta})$. Individual observations within sequences will be identified with a subscript, e.g. $\mathbf{y}_i$ will indicate element i in the sequence $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$, and *iid* draws from the model will be denoted using superscripts, e.g. $\mathbf{x}^{(r)} \overset{iid}{\sim} p(\cdot \mid \boldsymbol{\theta}), r = 1, \ldots, R$, will indicate $R \geq 1$ *iid* draws $\mathbf{x}^{(r)}$ from the model. Additional chapter-specific notation will also be clarified at various points throughout this thesis.

1.7 Recurring benchmark models

Our goal for this thesis is to develop methodology for performing likelihood-free Bayesian parameter inference for dynamic, stochastic simulation models, with a focus on the sort of models that appear in economics and the social sciences. To demonstrate the empirical performance of such methodology, we will conduct experiments in which we attempt to recover ground-truth parameter posteriors (or approximations of which through Monte Carlo techniques such as MCMC) using a series of different methods. For this reason, we will introduce in this section a collection of dynamic, stochastic simulation models that will appear at multiple points throughout the thesis during different benchmarking experiments. (Additional models that appear only once in this thesis will be introduced where they are used.)

Apart from one or two exceptions which we will highlight, the models we list below have been chosen according to the following criteria:

1. their prevalence as test cases within the simulation-based inference literature, both more broadly and within the specific context of economics and the social sciences, or the fact that they share similar features with such models; and

2. analytic or numerical tractability of the likelihood function and/or posterior density over parameters.

The latter criterion is critical to the proper and thorough benchmarking of different simulation-based Bayesian inference procedures: to properly and clearly assess the abilities of different methods to approximate the classical Bayesian posterior, it is typically necessary to use benchmark models whose structure allows for the actual classical Bayesian posterior to be obtained or estimated accurately using approximate techniques that target the exact posterior, such as MCMC.

A possible criticism of this selection criteria is that it can be difficult to trust that any good performance seen by such methods only in experiments performed with simpler models will extend to more complex models, such as large-scale agent-based models that may be encountered in the social sciences. While we agree that it is reasonable to retain a degree of suspicion regarding the ability of any methodology's good performance to extend to more complex cases, we submit that we are better off assessing their performance on simpler cases than to forgo this step altogether, since there are few – if any – alternative approaches to assessing the quality of approximate Bayesian inference procedures of which we are aware. Furthermore, if a method fails to provide a reasonable posterior approximation in simple cases, we argue that it would be difficult to trust that the performance would *improve* with the complexity.

1.7.1 Ricker model

The Ricker model is a simple model of ecological dynamics that exhibits chaotic behaviour and has an intractable likelihood function. The state of the model, which tracks the size $N_t \in \mathbb{R}_{\geq 0}$ of a population over discrete time steps $t = 1, \ldots, n$, evolves as

$$\log N_{t+1} = \log r + \log N_t - N_t + \sigma \epsilon_t, \tag{1.50}$$

where $r > 0$ is a growth parameter and $\epsilon_t \sim \mathcal{N}(0, 1)$. Following Wood (2010), we assume Poissonian observations

$$\mathbf{y}_t \sim \mathrm{Po}(\phi N_t) \in \mathbb{N}, \tag{1.51}$$

where $\phi > 0$ is a scale parameter. At various points in this thesis, we assume the task of recovering the posterior distribution for $\boldsymbol{\theta} = (\log r, \phi, \sigma)$ given a time series of length

$n = 50$, $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n) \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*)$ with $\boldsymbol{\theta}^* = (4, 10, 0.3)$. We take $N_0 = 1$. We further assume the following independent, uniform priors for each parameter:

$$\log r \sim \mathcal{U}(3, 8), \tag{1.52}$$

$$\phi \sim \mathcal{U}(0, 20), \tag{1.53}$$

$$\sigma \sim \mathcal{U}(0, 0.6). \tag{1.54}$$

To obtain samples from the ground truth posterior of the Ricker model we employ particle Markov chain Monte Carlo (PMCMC) using a simple bootstrap particle filter. We follow the guidelines of Schmon et al. (2021), first estimating the posterior covariance in a shorter prior run and then tuning the random walk proposal as well as the particle filter. We run the algorithm for 2×10^5 iterations eventually retaining a thinned subset of 10^3 samples as our baseline.

1.7.2 Geometric Brownian motion

We consider two variants of a geometric Brownian motion model: a univariate case and multivariate case. These will be presented separately below.

1.7.2.1 Univariate geometric Brownian motion

Geometric Brownian motion (GBM) is a stochastic differential equation widely used in mathematical finance to model the dynamics of a stock price x_t evolving with time t according to

$$\mathrm{d}x_t = \mu x_t \mathrm{d}t + \sigma x_t \mathrm{d}W_t, \tag{1.55}$$

where μ is the percentage drift, σ is the volatility, and W_t is a Brownian motion. This model permits an exact discretisation with $i = 1, 2, \ldots, n-1$ as

$$\log \mathbf{x}_{i\Delta t} = \log \mathbf{x}_{(i-1)\Delta t} + \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}\,\epsilon_i, \tag{1.56}$$

which implicitly defines the model $p(\mathbf{x} \mid \boldsymbol{\theta})$ from which we simulate. For all simulations, we fix $\mathbf{x}_0 = 10$, $n = 100$, and $\Delta t = 1/(n-1)$, and simulate the dynamics over the interval $[0, 1]$, such that $T = 1$.

We consider the task of recovering the posterior for parameters $\boldsymbol{\theta} = (\mu, \sigma)$ given an observation $\mathbf{y} = (\mathbf{y}_0, \mathbf{y}_{\Delta t}, \mathbf{y}_{2\Delta t}, \ldots, \mathbf{y}_{(n-1)\Delta t}) \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*)$ with $\boldsymbol{\theta}^* = (0.2, 0.5)$. We

assume independent, uniform priors on the parameters as follows:

$$\mu \sim \mathcal{U}(-1, 1), \quad \sigma \sim \mathcal{U}(0.2, 2). \tag{1.57}$$

Inference is amenable to standard, exact likelihood-based Bayesian techniques such as MH sampling using the transition density implied by (1.56). We use this fact to obtain an approximate ground-truth posterior using MH.

1.7.2.2 Multivariate geometric Brownian motion

We also consider a multivariate geometric Brownian motion (MVGBM), which is used in a variety of applications in financial time-series modelling and an extension of the univariate case considered above. The model is a stochastic differential equation in which each of the d components, labelled $i \in \{1, 2, \ldots, d\}$, of the path $\mathbf{X}_t = \left(X_t^1, X_t^2, \ldots, X_t^d\right) \in \mathbb{R}_+^d$ evolve according to

$$\mathrm{d}X_t^i = X_t^i \left(\left[b_i - \frac{1}{2} \sum_{j=1}^{d} \sigma_{ij}^2 \right] \mathrm{d}t + \sum_{j=1}^{d} \sigma_{ij} \, \mathrm{d}W_t^j \right), \tag{1.58}$$

where the b_i are drift coefficients, the σ_{ij} are volatility coefficients, and W_t^i is a Brownian motion.

We consider the case of $d = 3$ and the task of estimating the posterior for the parameters $\boldsymbol{\theta} = (b_1, b_2, b_3)$ given an observation $\mathbf{y} \sim p\left(\mathbf{x} \mid \boldsymbol{\theta}^*\right)$ of $T = 100$ points spaced equally with spacing $\Delta t = 1/(T-1)$, where $\boldsymbol{\theta}^* = (0.2, -0.5, 0.0)$. We take priors $b_i \sim \mathcal{U}(-1, 1)$ for each $i = 1, 2, 3$. This model once again permits both exact simulations and samples from the exact posterior since the transition density is once again tractable[7] and can be written as

$$\mathbf{x}_{t+\Delta t} \sim \mathcal{N}\left(\mathbf{x}_t + (\boldsymbol{\theta} - \boldsymbol{\gamma})\,\Delta t, \ \sigma\sigma^T \Delta t\right), \tag{1.59}$$

where

$$\boldsymbol{\gamma} = \frac{1}{2} \left[\sum_{j=1}^{d} \sigma_{1j}^2, \ \sum_{j=1}^{d} \sigma_{2j}^2, \ \sum_{j=1}^{d} \sigma_{3j}^2 \right]'. \tag{1.60}$$

In our experiments, we take

$$\sigma = \begin{pmatrix} 0.5 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.3 \\ 0.0 & 0.0 & 0.2 \end{pmatrix}. \tag{1.61}$$

[7] Assuming that σ is of full rank.

1.7.3 The Brock & Hommes agent-based model

We consider a variant of the model proposed by (Brock and Hommes, 1998), which has previously been used in ABM calibration experiments (Platt, 2020). The model dynamics can be expressed as the following system of coupled equations:

$$\mathbf{x}_{t+1} = \frac{1}{R}\left[\sum_{h=1}^{H} n_{h,t+1}\left(g_h \mathbf{x}_t + b_h\right) + \epsilon_{t+1}\right], \ \epsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{1.62}$$

$$n_{h,t+1} = \frac{\exp\left(\beta U_{h,t}\right)}{\sum_{h'=1}^{H} \exp\left(\beta U_{h',t}\right)}, \tag{1.63}$$

$$U_{h,t} = \left(\mathbf{x}_t - R\mathbf{x}_{t-1}\right)\left(g_h \mathbf{x}_{t-2} + b_h - R\mathbf{x}_{t-1}\right), \tag{1.64}$$

where R, β, σ are parameters. We follow Platt (2020) and assume that $H = 4, R = 1.0, \sigma = 0.04, g_1 = b_1 = b_4 = 0$ and $g_4 = 1.01$. β will be 120 or 10, depending on the experiment, and we note below which value is used. We consider the task of estimating the posterior $\pi\left(\boldsymbol{\theta} \mid \mathbf{y}\right)$, where $\boldsymbol{\theta} = (g_2, b_2, g_3, b_3)$, $\mathbf{y} := (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T) \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*)$ is the pseudo-observation, $T = 100$, and $\boldsymbol{\theta}^*$ is the parameter setting used to generate $\mathbf{y}$.

We note that by rewriting the above system of equations, we are able to find the transition density for observation $\mathbf{y}_{t+1}$ as

$$p(\mathbf{y}_{t+1} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{y}_{t+1}; f(\mathbf{y}_{t-2:t}, \boldsymbol{\theta}), \frac{\sigma^2}{R^2}\right) \tag{1.65}$$

where

$$f\left(\mathbf{y}_{t-2:t}, \boldsymbol{\theta}\right) = \frac{1}{R}\sum_{h=1}^{H} \frac{\exp\left[\beta\left(\mathbf{y}_t - R\mathbf{y}_{t-1}\right)\left(g_h \mathbf{y}_{t-2} + b_h - R\mathbf{y}_{t-1}\right)\right]}{\sum_{h'=1}^{H} \exp\left[\beta\left(\mathbf{y}_t - R\mathbf{y}_{t-1}\right)\left(g_{h'} \mathbf{y}_{t-2} + b_{h'} - R\mathbf{y}_{t-1}\right)\right]}\left(g_h \mathbf{y}_t + b_h\right). \tag{1.66}$$

In this way, we are able to obtain approximate ground truth posteriors with standard MCMC techniques such as MH.

1.7.4 Generalised stochastic epidemics

A generalised stochastic epidemic (GSE) model (Kypraios, 2007) simulates the spread of an infection through a fixed population of N individuals. Individuals are initially susceptible, may become infected, and subsequently recover without the possibility of reinfection. Dynamics of the model are determined by parameters β and γ, which

control the rate of infection and recovery according to the following transition probabilities:

$$P\left[X_{t+\delta t} - X_t = -1, Y_{t+\delta t} - Y_t = 1 \mid \mathcal{H}_t\right] = \beta X_t Y_t \delta t + o(\delta t), \tag{1.67}$$

$$P\left[X_{t+\delta t} - X_t = 0, Y_{t+\delta t} - Y_t = -1 \mid \mathcal{H}_t\right] = \gamma Y_t \delta t + o(\delta t), \tag{1.68}$$

$$P\left[X_{t+\delta t} - X_t = 0, Y_{t+\delta t} - Y_t = 0 \mid \mathcal{H}_t\right] = 1 - \beta\, X_t Y_t \delta t + \gamma Y_t \delta t + o(\delta t), \tag{1.69}$$

where X_t and Y_t are the number of susceptible and infected individuals at time $t \in [0, T]$, respectively, and $\mathcal{H}_t$ is a sigma-algebra generated by the process up until time t. These three transition probabilities thus capture infection, recovery, and an absence of activity, respectively.

We consider the problem of recovering the posterior density for $\boldsymbol{\theta} = (\beta, \gamma)$ given observations of the infections and recoveries occurring in the observation period $[0, T]$ with $T = 50$ in a system of $Z = 100$ individuals. For every simulation, the epidemic begins with one infected individual at time $t = 0$. We generate "empirical" data at parameters $\boldsymbol{\theta}^* = (10^{-2}, 10^{-1})$, and assume Gamma priors for both β and γ,

$$\beta \sim \Gamma(\lambda_\beta, \nu_\beta), \tag{1.70}$$

$$\gamma \sim \Gamma(\lambda_\gamma, \nu_\gamma), \tag{1.71}$$

with $\lambda_\beta = 0.1$, $\nu_\beta = 2$, $\lambda_\gamma = 0.2$, and $\nu_\gamma = 0.5$. It can be shown (Kypraios, 2007) that this prior is conjugate for the model, leading to the posterior density

$$\pi(\beta, \gamma \mid \mathbf{I}, \mathbf{R}) \propto \beta^{\lambda_\beta + n_I - 2} \exp\left\{-\beta\left(\int_{\phi_1}^{T} X_t Y_t \,\mathrm{d}t + \nu_\beta\right)\right\} \gamma^{\lambda_\gamma + n_R - 1} \exp\left\{-\gamma\left(\int_{\phi_1}^{T} Y_t \,\mathrm{d}t + \nu_\gamma\right)\right\}, \tag{1.72}$$

where $\mathbf{I}$ and $\mathbf{R}$ are the infection and recovery times, respectively, n_I and n_R are the total number of individuals in the model that are infected and that recover over the course of the simulation, respectively, and ϕ_1 is the time of the first infection. Thus, samples can be drawn from the exact posterior for a given dataset simulated by this model. We simulate the model using the Gillespie algorithm (Gillespie, 1977), such that the lengths of the simulated sequences, and the spacing between points in the sequences, are both also random.

1.7.5 Ornstein-Uhlenbeck process

The Ornstein-Uhlenbeck (OU) process (Uhlenbeck and Ornstein, 1930) is a prototypical Gauss–Markov stochastic differential equation (SDE) model. We discretise the SDE such that the data $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T), \mathbf{x}_i \in \mathbb{R}$ is generated according to

$$\mathbf{x}_i = \theta_1 \exp(\theta_2)\Delta t + (1 - \theta_1 \Delta t)\mathbf{x}_{i-1} + \frac{\epsilon_i}{2},$$

where $\Delta t = 0.2$ is the time discretisation, $\boldsymbol{\theta} = (\theta_1, \theta_2)$ are the model parameters to be inferred, $T = 50$, and $\epsilon_i \sim \mathcal{N}(0, \Delta t)$. We generate $\mathbf{x}^* \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*)$ with $\boldsymbol{\theta}^* = (0.5, 1)$ and consider the task of estimating $\pi(\boldsymbol{\theta} \mid \mathbf{x}^*)$ given priors $\theta_1 \sim \mathcal{U}(0, 1)$ and $\theta_2 \sim \mathcal{U}(-2, 2)$.

Part II

Approximate Bayesian Inference with Path Signatures

Chapter 2

Path Signatures[1]

2.1 Introduction

Our goal in this thesis is to develop likelihood-free approaches to Bayesian parameter inference for stochastic time-series simulation models. For this purpose, we have investigated the use of path signatures – an infinite-dimensional object arising in stochastic analysis and the theory of controlled differential equations that describes the geometry of multidimensional paths – as a natural feature set for sequential data. In this chapter, we provide an overview of path signatures, and how they may be used in practical inference and learning settings. This will form the basis of our own investigations into its use in LFI as presented in Chapters 3 through 5.

2.2 Path signatures

2.2.1 Path integrals and signatures

Let $\mathcal{H}$ be a Hilbert space and $h : [0, T] \to \mathcal{H}$ be a $\mathcal{H}$-valued path on interval $[0, T]$. For $p \geq 1$, we denote the p-variation of h over the interval $[s, t] \subseteq [0, T]$ as

$$\|h\|_{p-\mathrm{var},[s,t]} := \left(\sup_{\zeta(s,t)} \sum_{i=1}^{n-1} \left\| h_{t_{i+1}} - h_{t_i} \right\|_{\mathcal{H}}^{p} \right)^{1/p}$$

[1]This chapter is based on the introductory sections to Dyer et al. (2022c) and from a later, unreleased draft of Dyer et al. (2021a), both of which are joint work with Patrick Cannon and Sebastian M. Schmon.

where the supremum is taken over all finite partitions $\zeta(s,t)$ of the domain and $n = |\zeta(s,t)|$. Throughout this work, we will primarily consider $\mathcal{H}$-valued paths of bounded variation over the entire interval $[0,T]$, i.e. paths of finite p-variation for $p = 1$ such that

$$\|h\|_{1-\text{var}} := \sup_{\zeta(0,T)} \sum_{i=1}^{n-1} \left\|h_{t_{i+1}} - h_{t_i}\right\|_{\mathcal{H}} < \infty,$$

where the interval $[0,T]$ is omitted from the subscript for simplicity. We denote with $BV([0,T],\mathcal{H})$ the space of all such paths. The *path signature* (see e.g. Lyons et al., 2007) of h, denoted $\text{Sig}(h)$, maps such paths to an infinite series of tensors:

$$\text{Sig} : BV([0,T],\mathcal{H}) \to \prod_{m\geq 0} \mathcal{H}^{\otimes m}, \quad h \mapsto (1, S_1(h), S_2(h), \dots), \tag{2.1}$$

where

$$\prod_{m\geq 0} \mathcal{H}^{\otimes m} := \mathbb{R} \oplus \mathcal{H} \oplus (\mathcal{H}\otimes\mathcal{H}) \oplus \cdots \oplus \mathcal{H}^{\otimes m} \oplus \dots \tag{2.2}$$

and where we define recursively

$$S_m := \int_0^T \mathrm{d}h^{\otimes m} := \int_0^T \int_0^t \mathrm{d}h^{\otimes(m-1)} \otimes \mathrm{d}h_t. \tag{2.3}$$

In the above, we have adopted the convention that $\mathcal{H}^{\otimes 0} = \mathbb{R}$.

Example 1 (Example 2.3, Kiraly and Oberhauser (2019))**.** *Let h_t take values in $\mathbb{R}^2$, $h_t = (a_t, b_t)$. Then*

$$S_1(h) = \begin{bmatrix} \int_0^T \mathrm{d}a_t \\ \int_0^T \mathrm{d}b_t \end{bmatrix} \quad \textit{and} \quad S_2(h) = \begin{bmatrix} \int_0^T \int_0^{t_2} \mathrm{d}a_{t_1}\mathrm{d}a_{t_2} & \int_0^T \int_0^{t_2} \mathrm{d}a_{t_1}\mathrm{d}b_{t_2} \\ \int_0^T \int_0^{t_2} \mathrm{d}b_{t_1}\mathrm{d}a_{t_2} & \int_0^T \int_0^{t_2} \mathrm{d}b_{t_1}\mathrm{d}b_{t_2} \end{bmatrix}.$$

These terms can be further interpreted geometrically: the terms in $S_1(h)$ capture the increments along each dimension, while the off-diagonal elements of $S_2(h)$ capture the areas above and below the curve; see Figure 2.1. Higher order terms capture higher order notions of area that are more difficult to visualise and interpret.

Remark 1. *Since we have assumed our paths to be of bounded variation, the integrals above can be understood as the Riemann-Stieljes integrals with respect to h. When the underlying path is not smooth, the integrals are taken to be stochastic or rough path integrals (Chevyrev and Oberhauser, 2018). For example, in the case of Brownian motion in $\mathbb{R}^d$, the integrals are stochastic and can be taken in the Stratonovich sense. For a larger class of stochastic processes, rough path theory (Lyons et al.,*

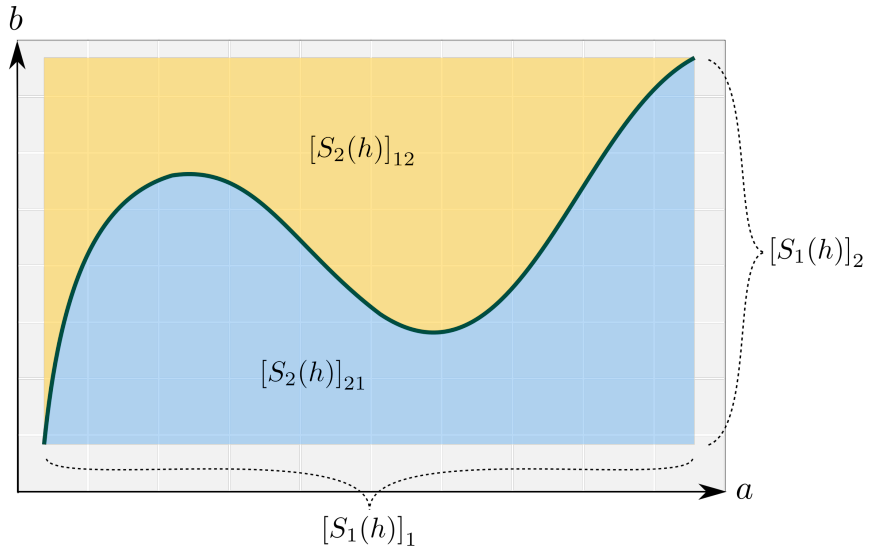


Figure 2.1: Geometric interpretation of the signature terms for an example two-dimensional path, shown as the dark green curve. Depth-1 terms correspond to the increments $a_T - a_0$ and $b_T - b_0$, while the depth-2 terms $[S_2(h)]_{21}$ and $[S_2(h)]_{12}$ correspond to the blue and yellow areas, respectively.

2007) provides an integration theory that enables the computation of the terms in the signature. As we will discuss later, this work considers throughout only linear interpolations between points in time series, so all paths considered here are of finite variation.

Path signatures are thus infinite sequences of statistics for path-valued random variables capturing information regarding the order of observations along, and the interaction between, different channels of the path. They are grounded in the theory of controlled differential equations (CDEs) and stochastic analysis, and appear in the solutions of CDEs and SDEs as obtained through a procedure analogous to Picard iterations for ordinary differential equations.

To see this, we follow Lyons et al. (2007) and let V and W be two Banach spaces, $B : V \to \mathbf{L}(W, W)$ be a bounded linear map – where $\mathbf{L}(W, W)$ denotes the space of bounded linear mappings from $W \to W$ – and $h : [0, T] \to V$ be a continuous path of bounded variation. Consider the following set of linear equations:

$$\mathrm{d}g_t = Bg_t \,\mathrm{d}h_t, \quad g_0 \in W \tag{2.4}$$

$$\mathrm{d}\phi_t = B\phi_t \,\mathrm{d}h_t, \quad \phi_0 \in \mathbf{L}(W, W). \tag{2.5}$$

Here, $Bg_t\,\mathrm{d}h_t$ is taken to mean $[B(\mathrm{d}h_t)]\,(g_t)$ while $B\phi_t\,\mathrm{d}h_t$ is $B(\mathrm{d}h_t)\circ\phi_t$. By applying the aforementioned iterative procedure to recover the solution ϕ_t to (2.5), we obtain

$$\phi_t = \sum_{m\geq 0} B^{\otimes m} \int_0^t \mathrm{d}h^{\otimes m}, \tag{2.6}$$

in which we see that the signature terms, Equation (2.3), appear in the summand. The solution to (2.4) is then obtained from the flow ϕ_t as $g_t = \phi_t(h_0)$. Similarly, a solution to the following linear SDE driven by Brownian motion W,

$$\mathrm{d}Y_t = A(Y_t)\circ\mathrm{d}W_t, \quad Y_0 = y_0$$

for some linear operator A, can be obtained as

$$Y_t = \sum_{m\geq 0} A^{\otimes m} S_{m,[0,t]}(W)\, y_0,$$

where $S_{m,[0,t]}(W)$ is the order-m tensor in the signature of W_t over interval $[0,t]$ and the integrals are taken in the Stratonovich sense (Lyons et al., 2007, Section 3.3.2). As we have seen here, signatures arise naturally as good approximations to solutions of CDEs and SDEs, and accurately describe the response of systems such as that of Equations (2.4)-(2.5) to an input signal h, where the inclusion of terms of increasing order further refine the approximate solution. The above sums, such as in Equation (2.6), converges as a result of the factorial rate of decay of the terms in the signature:

Proposition 3 (Proposition 2.2, Lyons et al. (2007))**.** *Let V be a Banach space and $h \in BV([0,T], V)$. Then, for each $m \geq 0$,*

$$\left\| \int_0^T \mathrm{d}h^{\otimes m} \right\|_{V^{\otimes m}} \leq \frac{\|h\|_{1-\mathrm{var}}}{m!}. \tag{2.7}$$

Remark 2. *The signature of a univariate path consists only of powers of the difference between the final and initial points in the stream (see e.g. Chevyrev and Kormilitzin, 2016, Example 5). Therefore in practice one always considers paths in at least two dimensions. This can always be achieved by including the observation time as a channel in the path.*

2.2.2 Key properties of path signatures

Signatures have a number of desirable properties. In the following subsections, we consider some of the main properties that we will make use of throughout this work.

2.2.2.1 Universal nonlinearity

One such property is *universal nonlinearity*: the signature captures all possible nonlinearities in path-valued random variables, in the sense that it is possible to approximate any nonlinear function of a path arbitrarily well with a linear functional of the signature. This is a consequence of the *shuffle product* property of signatures:

Theorem 1 (Theorem 2.29, Lyons et al. (2007)). *Let $h \in BV([0,T],\mathcal{H})$. Then*

$$\int_0^T \mathrm{d}h^{\otimes m} \otimes \int_0^T \mathrm{d}h^{\otimes m'} = \sum_{\sigma} \sigma\left(\int_0^T \mathrm{d}h^{\otimes(m+m')}\right),$$

where the sum is taken over all order shuffles, defined as

$$\{\sigma : \sigma \text{ is a permutation of } \{1,\dots,m+m'\} \text{ with } \sigma(1)<\cdots<\sigma(m), \sigma(m+1)<\cdots<\sigma(m+m')\}.$$

σ then acts on $\mathcal{H}^{\otimes(m+m')}$ as $\sigma(e_{i_1}\otimes\cdots\otimes e_{i_{m+m'}}) = e_{\sigma(i_1)}\otimes\cdots\otimes e_{\sigma(i_{m+m'})}$.

Applying the classical Stone-Weierstrass theorem[2] results in the stated universal nonlinearity property, which can be formalised as follows:

Theorem 2. *Let $\mathcal{K}$ be a compact set of non-tree-like[3] paths of bounded variation, and $C(\mathcal{K},\mathbb{R})$ be the space of continuous, real-valued function on $\mathcal{K}$. Then the space of linear functionals on signatures of paths in $\mathcal{K}$ is dense in $C(\mathcal{K},\mathbb{R})$; that is, for any $f \in C(\mathcal{K},\mathbb{R})$ and any $\varepsilon > 0$, there exists an $L \in \bigoplus_{m\geq 0}\mathcal{H}^{\otimes m}$ such that*

$$\sup_{h\in\mathcal{K}} \left| f(h) - L\left[\mathrm{Sig}(h)\right] \right| < \varepsilon.$$

2.2.2.2 Invariance properties

Further properties of the signature include its translation and reparameterisation invariance:

[2] An issue that arises in the application of the classical Stone-Weierstrass theorem in this context is that the space of interest to us – $BV([0,T],\mathcal{H})$ – is not locally compact. The classical Stone-Weierstrass theorem therefore cannot strictly be applied here. However, Chevyrev and Oberhauser (2018) demonstrate that a Stone-Weierstrass result exists by equipping the space of continuous bounded real-valued functions on $BV([0,T],\mathcal{H})$ with an appropriate topology. See Chevyrev and Oberhauser (2018) for details.

[3] See Section 2.2.2.2.

Proposition 4. *Let* $h \in BV([0,T],\mathcal{H})$, $a \in \mathcal{H}$, *and* $\psi : [0,T] \to [0,T]$. *Then* $\mathrm{Sig}(h+a) = \mathrm{Sig}(h)$ *and* $\mathrm{Sig}(h \circ \psi) = \mathrm{Sig}(h)$.

In this way, signatures are able to factor out nuisance and potentially infinite-dimensional symmetries where this is beneficial. However, when such invariances are disadvantageous, they can easily be destroyed with two extremely simple preprocessing techniques: *time-augmentation*, in which the path (t, h_t) is instead considered, and *basepoint augmentation*, in which $h_0 = c$ for some fixed constant $c \in \mathcal{H}$ is enforced for all paths under consideration.

A third, more interesting invariance property results from the signature's inability to identify regions of the path in which, informally speaking, a retracing of the path occurs (Chen, 1958; Hambly and Lyons, 2010; Boedihardjo et al., 2016); that is, for example, paths of the form $a \star b \star \overleftarrow{b} \star c$ for $a, b, c \in BV([0,T],\mathcal{H})$, where $\star$ denotes concatenation and $\overleftarrow{b}$ is the path b "run-backwards". Paths in which such retracings occur are referred to as *tree-like equivalent* to their reduced paths such that, for example, $a \star b \star \overleftarrow{b} \star c \sim_t a \star c$, where $\sim_t$ denotes tree-like equivalence. This phenomenon was originally studied in Chen (1958) for piecewise regular paths, and subsequently extended in Hambly and Lyons (2010) to paths of bounded variation in finite dimensional spaces. The most general form of this invariance property is provided by Boedihardjo et al. (2016), a special case of which may be stated as follows:

Theorem 3 (Boedihardjo et al. (2016))**.** *Let* V *be a Banach space and* $h, g \in BV([0,T],V)$. *Then* $\mathrm{Sig}(h) = \mathrm{Sig}(g)$ *iff* $h \sim_t g$.

In the real world, however, tree-like equivalent paths are rare and can straightforwardly be avoided by considering only time-augmented paths $h : [0,T] \to \mathcal{H} \times [0,T]$, $t \mapsto (t, h_t)$. Such a transformation ensures that the path is injective, meaning no partial retracing can occur at any point along the path. This, along with their universal nonlinearity property, demonstrates that signatures are powerful and faithful representations of paths and are, essentially, an injective feature map for path-valued random variables. Signatures are therefore an appealing option for performing inference for dynamic, stochastic processes.

2.2.3 The signature kernel

Computing iterated integrals for high- or potentially infinite-dimensional paths quickly becomes computationally infeasible due to the combinatorial explosion of terms in the signature with increasing depth. In part due to this, recent research effort (Kiraly and Oberhauser, 2019; Salvi et al., 2021) has been directed towards kernelising the feature map in Equation (2.1), permitting the use of the signature in learning procedures without explicit evaluation of the signature terms themselves. We provide here further details on the resultant *signature kernel*, of which we make use throughout the current work.

We follow Kiraly and Oberhauser (2019) and begin by defining the following for $A, B \in \prod_{m\geq 0} \mathcal{H}^{\otimes m}$:

$$A + B := (a_0 + b_0, a_1 + b_1, \dots), \tag{2.8}$$

and an inner product

$$\langle A, B \rangle := \sum_{m\geq 0} \langle a_m, b_m \rangle_{\mathcal{H}^{\otimes m}}, \tag{2.9}$$

where $A = (a_0, a_1, \dots)$, $B = (b_0, b_1, \dots)$, and

$$\langle u_1 \otimes \cdots \otimes u_m, v_1 \otimes \cdots \otimes v_m \rangle_{\mathcal{H}^{\otimes m}} = \prod_{j=1}^{m} \langle u_j, v_j \rangle_{\mathcal{H}}. \tag{2.10}$$

This leads us to the following norm on $\prod_{m\geq 0} \mathcal{H}^{\otimes m}$:

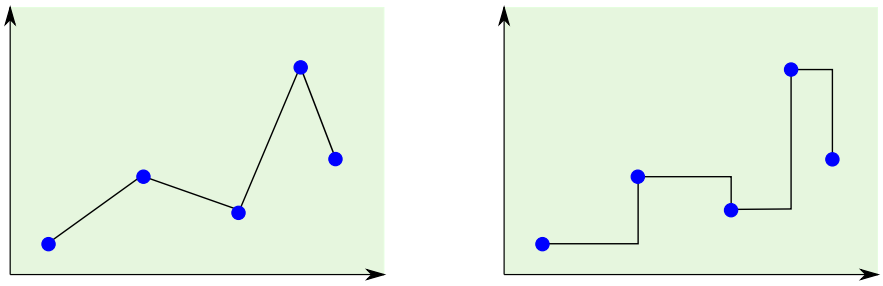
$$\|A\| := \sqrt{\sum_{m\geq 0} \|a_m\|^2_{\mathcal{H}^{\otimes m}}}. \tag{2.11}$$

Using the inner product (2.9) and the fact that $\mathrm{Sig}(h) \in \prod_{m\geq 0} \mathcal{H}^{\otimes m}$ for $h \in BV([0,T], \mathcal{H})$, we arrive at the definition of the signature kernel:

Definition 3 (Signature kernel, Kiraly and Oberhauser (2019))**.** *The* signature kernel *for $h, g \in BV([0,T], \mathcal{H})$ is*

$$k : BV([0,T], \mathcal{H}) \times BV([0,T], \mathcal{H}) \to \mathbb{R}, \quad (h, g) \mapsto \langle \mathrm{Sig}(h), \mathrm{Sig}(g) \rangle, \tag{2.12}$$

where the inner product is defined as in Equation (2.9).



(a) Linear interpolation (b) Rectilinear interpolation

Figure 2.2: Two interpolation schemes to convert a series of (blue) points into paths.

A key insight of Kiraly and Oberhauser (2019) was to recognise that evaluation of the signature kernel – which operates on *paths* in $\mathcal{H}$ – can be performed using only evaluations of an inner product κ that operates on *points* in the path, amounting to a kernel trick for the signature kernel. Kiraly and Oberhauser (2019) further describe an efficient Horner scheme to evaluate a truncated signature kernel that approximates Equation (2.12). In more recent work, Salvi et al. (2021) provide an alternative approach to approximating Equation (2.12) without truncation by observing that the signature kernel solves a Goursat partial differential equation. The solution to this Goursat problem may be obtained numerically with standard finite element methods, and can similarly be computed using only evaluations of an inner product κ on points in the path.

2.2.4 Path signatures in practice

In light of their interesting and useful properties described above, signatures can be seen as a canonical feature transformation for path-valued random variables. However, there exists an incongruity between our discussion so far and the scenarios faced in real-world settings: in reality and from the output of simulation models, we observe discretely sampled data $\mathbf{x} = (\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_n})$ at times $0 = t_1 < t_2 < \cdots < t_n = T$, where $\mathbf{x}_t \in \mathcal{X}$ for some finite-dimensional space $\mathcal{X}$ (for example $\mathbb{R}^d$ or $\mathbb{R}^{d \times d}$ for some $d \geq 1$), rather than continuous paths $x \in BV([0, T], \mathcal{H})$. This is dealt with naturally in the signature (kernel) literature in the following ways:

1. As noted by Kiraly and Oberhauser (2019), the aforementioned signature kernel trick can be used to introduce nonlinearities and embed the $\mathcal{X}$-valued sequence $\mathbf{x}$ in a Hilbert space. In particular, by choosing a reproducing kernel

$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with RKHS $\mathcal{H}$ and canonical feature map $\kappa(\mathbf{x}_t, \cdot) \in \mathcal{H}$ as the inner product on the data space $\mathcal{X}$, we may implicitly construct a sequence $(\kappa(\mathbf{x}_{t_1}, \cdot), \kappa(\mathbf{x}_{t_2}, \cdot), \ldots, \kappa(\mathbf{x}_{t_n}, \cdot))$ of points in $\mathcal{H}$ from sequences of data in $\mathcal{X}$.

2. To construct continuous paths from the discrete sequence above, an interpolation scheme is employed. While many interpolation schemes are possible, for example rectilinear interpolation (see Figure 2.2), the most common is linear interpolation. Indeed, Kiraly and Oberhauser (2019) and Salvi et al. (2021) assume a linear interpolation to construct *discretised* signature kernels operating on sequences of points, and we use this interpolation scheme throughout this work.

Remark 3. *The question of whether a linear interpolation is a good model for the evolution of the path between observations arises here. When the phenomenon and simulator is fundamentally discrete, the interpolation does not necessarily have any physical meaning and can be seen as an intermediate computational "trick" that gives us access to signature-based methods and other continuous-time approaches to performing inference with and modelling sequential data. When the system evolves and the model is expressed in continuous time, however, a linear interpolation may be a bad model for the evolution of the system in between observations, depending on the nature of the system and model. Intuitively, this problem may become more salient when the path is sampled at frequencies that are far smaller than those corresponding to the most rapid timescales of the system of interest. Linear interpolations are nonetheless common in applications of signatures since it (a) embodies, loosely speaking, a non-informative prior on the dynamics of the system between observations, and (b) significantly simplifies numerical evaluations of the signature through e.g. Chen's identity (see Remark 6, Appendix A; and Chen (1958)) and through Theorem 2 of Kiraly and Oberhauser (2019). Developing methods to account for the uncertainty in the intervening dynamics, or to impose stronger and more informative priors on the dynamics between observations, would however be an interest avenue for future work.*

By combining the above two steps, we may progress from a sequence $\mathbf{x}$ of points in $\mathcal{X}$ to a $\mathcal{H}$-valued, piecewise linear path h, given by

$$h_t := \kappa(\mathbf{x}_{t_i}, \cdot) + \frac{t - t_i}{t_{i+1} - t_i} \left(\kappa(\mathbf{x}_{t_{i+1}}, \cdot) - \kappa(\mathbf{x}_{t_i}, \cdot)\right) \text{ for } t \in [t_i, t_{i+1}], \ i = 1, \ldots, n-1. \tag{2.13}$$

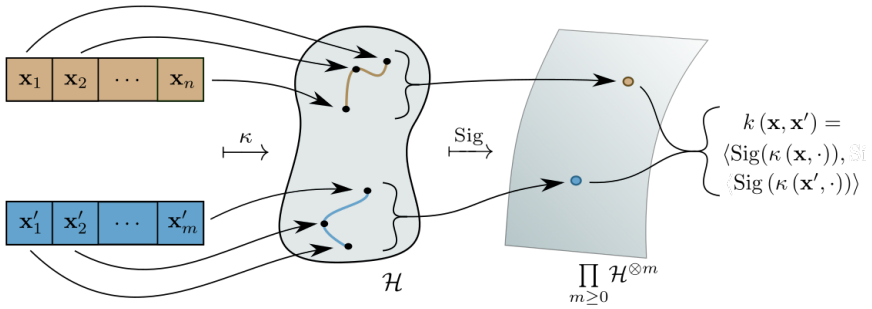


Figure 2.3: Time-series embedding via the signature kernel k with static kernel κ. The time-series $\mathbf{x}$, $\mathbf{x}'$ are lifted to paths in feature space $\mathcal{H}$, via κ and some interpolation scheme, before being mapped to a space of formal power series $\prod_{m\geq 0} \mathcal{H}^{\otimes m}$ of tensors via the signature.

Piecewise linear paths constructed in this way are naturally of bounded variation if, for example, κ is a uniformly bounded kernel[4]. We will assume this throughout, such that all observed sequences in $\mathcal{X}$ lift to piecewise linear paths of bounded variation in $\mathcal{H}$ under the feature map corresponding to κ, and denote the space of piecewise linear paths of bounded variation in $\mathcal{H}$ over time interval $[0, T]$ with $\mathcal{P}([0, T], \mathcal{H})$. We will furthermore abuse notation slightly by letting $\kappa(\mathbf{x}, \cdot) \in \mathcal{P}([0, T], \mathcal{H})$ denote the path in Equation (2.13), i.e. the linear interpolation of the lifted points $(\kappa(\mathbf{x}_{t_1}, \cdot), \kappa(\mathbf{x}_{t_2}, \cdot), \ldots, \kappa(\mathbf{x}_{t_n}, \cdot))$, while denoting the feature map for $\mathbf{x}_t$ with $\kappa(\mathbf{x}_t, \cdot) \in \mathcal{H}$. Finally, we will take $k(\mathbf{x}, \cdot) := \mathrm{Sig}(\mathbf{x})$ to mean the signature of the piecewise linear, $\mathcal{H}$-valued path $\kappa(\mathbf{x}, \cdot)$, while $\mathrm{Sig}(g)$ denotes the signature of a path $g \in BV([0, T], \mathcal{H})$. With this notation in place, we illustrate the way in which sequences are embedded with the signature kernel in Figure 2.3.

2.2.4.1 Further pre-processing

Prior to lifting the sequence to a path in $\mathcal{H}$, and depending on the nature of the data at hand, it is sometimes appropriate to apply a transformation to the data. The reason for doing so is that certain transformations may enable the signature to represent information in the stream more conveniently for the learning task at hand. A large set of such transformations have been proposed in the literature on inference using path signatures; see Morrill et al. (2020) for a recent summary and comparison of many of these. Here, we describe some of the most common pre-signature path transformations.

[4]See Proposition 7 below.

Cumulative sum Recall from Figure 2.1 that the depth 1 signature terms correspond to the increment along the path, and that a subset of the depth 2 terms correspond to the areas above and below the curve. For certain data types, for example non-negative binary or spiking data, the data may not be well-characterised by these terms by default. In such cases it can be beneficial to consider instead the cumulative sum of the observations (Kiŕaly and Oberhauser, 2019), which can intuitively be thought of as propagating information from earlier in the sequence to later in the stream, more readily exhibiting the structure of the stream. The effect of this can be to shift information into lower order terms in the signature, for example the increments (depth 1 terms).

Lead-lag transformation This transformation operates on a sequence $\mathbf{x} = (\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_n})$ as follows:

$$(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_n}) \mapsto \big((\mathbf{x}_{t_1}, \mathbf{x}_{t_1}), (\mathbf{x}_{t_1}, \mathbf{x}_{t_2}), (\mathbf{x}_{t_2}, \mathbf{x}_{t_2}), \ldots, (\mathbf{x}_{t_{n-1}}, \mathbf{x}_{t_n}), (\mathbf{x}_{t_n}, \mathbf{x}_{t_n})\big)\,. \tag{2.14}$$

Under this transformation, the number of channels in the sequence doubles, and the sequence length increases from n to $2n-1$. Applying this transformation enables the signature to emphasise certain properties of the path such as the quadratic variation and the Lévy area when combined with the cumulative sum (Gyurk, 2014; Chevyrev and Kormilitzin, 2016). For datasets for which these quantities are believed to be important, applying the lead-lag transformation may be appropriate.

Delay transformation A similar transformation to the above is a delay transformation, for example the lag-1 delay transformation:

$$(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_n}) \mapsto ((\mathbf{x}_{t_1}, \mathbf{x}_{t_2}), (\mathbf{x}_{t_2}, \mathbf{x}_{t_3}), \ldots, (\mathbf{x}_{t_{n-1}}, \mathbf{x}_{t_n})). \tag{2.15}$$

While the number of channels doubles here also, this transformation may be computationally preferable to the lead-lag transformation, since the length of the sequence does not increase in this case.

2.2.4.2 Augmentations

As noted previously, two augmentations can be applied to remove the signature's translation and reparameterisation invariance properties:

Time augmentation, in which the uniformly increasing time index $0 = t_1 < t_2 < \cdots < t_n = T$ is added as a channel in the sequence:

$$(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_n}) \mapsto ((t_1, \mathbf{x}_{t_1}), (t_2, \mathbf{x}_{t_2}), \ldots, (t_n, \mathbf{x}_{t_n})), \tag{2.16}$$

denoting the times at which the points in the series occurred.

Basepoint augmentation, in which all sequences are enforced to assume a common but otherwise arbitrary initial value. This can be achieved by simply concatenating an arbitrary constant value to the beginning of each sequence.

Chapter 3

Approximate Bayesian Computation with Path Signatures[1]

As described in Section 1.4.2, simulation models of scientific interest often lack a tractable likelihood function, which precludes standard likelihood-based statistical inference. Consequently, traditional approaches to statistical inference are infeasible and alternative LFI methods are usually adopted. One of the most widely used LFI methods is ABC (Tavaré et al., 1997; Pritchard et al., 1999; Beaumont et al., 2002), in which the Bayesian posterior distribution is approximated by sampling parameters $\boldsymbol{\theta}$ from a proposal distribution (e.g. the prior density) and synthetic datasets $\mathbf{x}$ from a stochastic simulator, with likelihood denoted $p(\mathbf{x} \mid \boldsymbol{\theta})$, and comparing the output $\mathbf{x}$ with real data $\mathbf{y}$. The simplest form of ABC then makes the following decision: if the simulator output is sufficiently 'close' to the observation, then $\boldsymbol{\theta}$ is retained as a sample from the approximate posterior distribution; otherwise it is discarded.

However, measuring closeness between model outputs is known to be challenging, particularly for time-series data, which can exhibit complex dependency structures and may be multivariate and sampled at irregular time intervals. A common approach is to attempt to distil important features of the data using summary statistics and compare these instead (see e.g. Prangle, 2018). In practice, informative summary statistics are difficult to craft: a consequence of the Pitman-Koopman-Darmois Theorem in Proposition 2 is that (low-dimensional) sufficient statistics are not generally available for arbitrary models, meaning that summarising generally introduces

[1]The contents of this chapter are drawn from a later, unreleased draft of Dyer et al. (2021a), which is joint work with Patrick Cannon and Sebastian M. Schmon.

a loss of information. This presents a trade off: a poor choice can materially bias the algorithm away from the true posterior distribution, yet constructing a sufficiently powerful choice can require substantial domain expertise, problem insight, and costly experimentation (see e.g. Drovandi and Frazier, 2021, for a recent comparison of methods with and without summaries).

In other approaches, the engineering of summary statistics is bypassed altogether (see e.g. Bernton et al., 2019; Park et al., 2016), though existing methods of this type are generally not suitable for time-series models without further adjustment. Some of the most popular approaches of this kind were reviewed in Section 1.5.1, where we saw that there are currently few, if any, ABC methods well-suited to time-series models, due to e.g. unrealistic assumptions such as *iid* data.

In this chapter, we present two novel methods for performing ABC for time-series models that bypass the difficult problem of manually constructing summary statistics. Our approach leverages so-called *path signatures*, a key object in the mathematics of rough path theory (see e.g. Lyons, 2014). Signatures have been employed successfully in a variety of tasks, from hand-gesture recognition (Li et al., 2017a) to the early identification of Alzheimer's disease (Moore et al., 2019), and constitute a natural feature set for multivariate and even irregularly sampled sequential data (Salvi et al., 2021). We demonstrate that the path signature can be employed directly as a summary statistic, or in the context of a semi-automatic projection approach, to construct a powerful distance measure for time-series data in ABC, and further that such an approach can recover more accurate posterior estimates than existing techniques.

3.1 Approximate Bayesian computation with signature transforms

In this section, we introduce the use of path signatures as a flexible and general framework for performing ABC for complex time-series models. Given its unique properties, the path signature and its associated kernel are natural candidates for feature maps and discrepancy measures in ABC to handle irregularly spaced and potentially multivariate time series data. In this section, we will introduce and investigate two simple but powerful techniques for incorporating signatures in ABC.

3.1.1 Signature ABC

Though signatures are infinite-dimensional objects, we can leverage their kernel representation (see Definition 3) to compute the distance between two sequences $\mathbf{x}, \mathbf{y}$ as the norm induced by the associated inner product. That is, for two time series $\mathbf{x}$ and $\mathbf{y}$, we can interpret the signature of their lifted paths as a *summary statistic*, $\mathbf{s}(\mathbf{x}) = \text{Sig}(\mathbf{x})$, and compute

$$\rho(\mathbf{s}(\mathbf{x}), \mathbf{s}(\mathbf{y})) := \|\text{Sig}(\mathbf{x}) - \text{Sig}(\mathbf{y})\|^2 = k(\mathbf{x}, \mathbf{x}) + k(\mathbf{y}, \mathbf{y}) - 2\,k(\mathbf{x}, \mathbf{y}), \tag{3.1}$$

where, again, $k(\mathbf{x}, \mathbf{y}) = \langle \text{Sig}(\mathbf{x}), \text{Sig}(\mathbf{y}) \rangle$. The resulting distance can be computed easily using the `sigkernel`[2] package or alternatives[3] and used to derive an ABC posterior via Equations (1.15)-(1.16). For example, it may be embedded either in rejection ABC, leading to the ABC posterior

$$\pi_{\text{REJ}}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \pi(\boldsymbol{\theta}) \int \mathbb{1}\left\{\|\text{Sig}(\mathbf{x}) - \text{Sig}(\mathbf{y})\|^2 \leq \varepsilon\right\} p(\mathbf{x} \mid \boldsymbol{\theta}) \mathrm{d}\mathbf{x},$$

or alternatively following the approach of Schmon et al. (2020) as a loss in the generalized approximate posterior (1.17), that is

$$\pi_{\text{GBI}}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \pi(\boldsymbol{\theta}) \int e^{-w\|\text{Sig}(\mathbf{x}) - \text{Sig}(\mathbf{y})\|^2} p(\mathbf{x} \mid \boldsymbol{\theta}) \mathrm{d}\mathbf{x}.$$

In both cases our method straightforwardly extends classical approaches by using the distance function (3.1), suggesting the name Signature ABC (S-ABC). In the latter case, Monte Carlo samples can be obtained using, for example, a pseudo-marginal approach (Beaumont, 2003; Andrieu et al., 2009). For the remainder of this chapter, however, we will only consider standard rejection ABC (REJ-ABC) in the interest of a simple and fair comparison with alternative distance measures.

We next consider the theoretical properties of the S-ABC posterior. In particular, we consider two asymptotic regimes: the correctness of the S-ABC posterior for fixed data and as the ABC tolerance hyperparameter $\varepsilon \to 0$; and the behaviour of the S-ABC posterior for fixed ε and as the number of samples $n \to \infty$ in the interval $[0, T]$ or, equivalently, as the sampling rate tends to infinity.

[2] https://github.com/crispitagorico/sigkernel
[3] See e.g. https://github.com/tgcsaba/KSig.

3.1.1.1 Behaviour as $\varepsilon \to 0$ for fixed n

We first demonstrate that the discrepancy measure in Equation (3.1) satisfies the conditions specified in Proposition 3.1 of Bernton et al. (2019), which gives a statement on the convergence of ABC posteriors to the true posterior under certain regularity conditions on the simulator's likelihood function as $\varepsilon \to 0$. A specific case of the statement is as follows:

Proposition 5 (Proposition 3.1, Bernton et al. (2019))**.** *Let $\mathcal{X} := \mathbb{R}^d$, $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n) \in \mathcal{X}^n$, and $\mathcal{D} : \mathcal{X}^n \times \mathcal{X}^n \to \mathbb{R}_{\geq 0}$ be a non-negative distance measure on $\mathcal{X}^n$. Suppose $p(\mathbf{x} \mid \boldsymbol{\theta})$ is the continuous density associated with simulated data $\mathbf{x} \in \mathcal{X}^n$ and that*

$$\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta} \setminus \mathcal{N}_{\boldsymbol{\Theta}}} p(\mathbf{x} \mid \boldsymbol{\theta}) < \infty,$$

where $\mathcal{N}_{\boldsymbol{\Theta}}$ is a set such that $\pi(\boldsymbol{\theta}) = 0 \, \forall \boldsymbol{\theta} \in \mathcal{N}_{\boldsymbol{\Theta}}$. Suppose further that there exists $\bar{\varepsilon} > 0$ such that

$$\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta} \setminus \mathcal{N}_{\boldsymbol{\Theta}}} \sup_{\mathbf{z} \in \mathcal{A}^{\bar{\varepsilon}}} p(\mathbf{z} \mid \boldsymbol{\theta}) < \infty,$$

where $\mathcal{A}^{\bar{\epsilon}} := \{\mathbf{z} : \mathcal{D}(\mathbf{y}, \mathbf{z}) \leq \bar{\varepsilon}\}$. Suppose that $\mathcal{D}$ is continuous. If $\mathcal{D}(\mathbf{y}, \mathbf{z}) = 0$ iff $\mathbf{y} = \mathbf{z}$ then, keeping $\mathbf{y}$ fixed, the ABC posterior converges strongly to the posterior as $\varepsilon \to 0$.

Therefore, provided that the stated regularity conditions on the simulator's likelihood function are met, showing that the distance function in Equation (3.1) is continuous and injective is sufficient to show that the S-ABC posterior converges to the true posterior as $\varepsilon \to 0$. These requirements are indeed met under the assumptions of Proposition 5 and under additional benign conditions:

Proposition 6. *Let $\mathcal{X} := \mathbb{R}^d$, $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n) \in \mathcal{X}^n$ be the fixed real-world dataset, and $\mathcal{D}(\mathbf{y}, \cdot)$ be as in Equation (3.1), i.e.*

$$\mathcal{D}(\mathbf{y}, \cdot) : \mathcal{X}^n \to \mathbb{R}_{\geq 0}, \quad \mathbf{x} \mapsto \|\mathrm{Sig}(\mathbf{y}) - \mathrm{Sig}(\mathbf{x})\|^2 .$$

Assume both $\mathbf{y}$ and $\mathbf{x}$ are time- and basepoint-augmented, and that $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a continuous, uniformly bounded, injective kernel. Then $\mathcal{D}(\mathbf{y}, \cdot)$ is uniformly continuous.

To prove this, we will proceed by noting that each constituent map in the above operation is a continuous map, and the result follows since compositions of continuous maps are continuous. First, we will show that the one-variation of a basepoint-augmented sequence/piecewise linear path is a norm:

Lemma 1. *Let $\mathcal{X}^n$ be the space of length-n basepoint-augmented sequences in $\mathcal{X} = \mathbb{R}^d$ and $\mathbf{x}, \mathbf{z} \in \mathcal{X}^n$. Then the one-variation*

$$\|\mathbf{x}\|_{1-\text{var}} = \sum_{i=1}^{n-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_{\mathcal{X}} \tag{3.2}$$

is a norm on $\mathcal{X}^n$.

Proof. The triangle inequality follows immediately as a result of the triangle inequality for the norm on $\mathcal{X}$:

$$\begin{aligned}
\|\mathbf{x} + \mathbf{z}\|_{1-\text{var}} &= \sum_{i=1}^{n-1} \|(\mathbf{x}_{i+1} + \mathbf{z}_{i+1}) - (\mathbf{x}_i + \mathbf{z}_i)\|_{\mathcal{X}} \\
&\leq \sum_{i=1}^{n-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_{\mathcal{X}} + \|\mathbf{z}_{i+1} - \mathbf{z}_i\|_{\mathcal{X}} \\
&= \|\mathbf{x}\|_{1-\text{var}} + \|\mathbf{z}\|_{1-\text{var}}.
\end{aligned}$$

Absolute homogeneity is also immediate:

$$\|s\mathbf{x}\|_{1-\text{var}} = \sum_{i=1}^{n-1} \|s\mathbf{x}_{i+1} - s\mathbf{x}_i\|_{\mathcal{X}} = |s| \sum_{i=1}^{n-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_{\mathcal{X}} = |s| \|\mathbf{x}\|_{1-\text{var}}.$$

Finally, since the streams are basepoint-augmented, meaning $\mathbf{x}_1 = 0$ for all $\mathbf{x} \in \mathcal{X}^n$, we have that $\|\mathbf{x}\|_{1-\text{var}} = 0$ iff $\mathbf{x} = (0, 0, \ldots, 0)$:

$$\|\mathbf{x}\|_{1-\text{var}} = 0 \implies \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_{\mathcal{X}} = 0 \; \forall \; i = 1, \ldots, n-1 \implies \mathbf{x}_i = \mathbf{x}_1 = 0 \; \forall \; i.$$

□

We next show that lifting length-n basepoint-augmented sequences in $\mathcal{X}$ to sequences in $\mathcal{H}$ is continuous if the canonical feature map ϕ associated with κ is itself continuous:

Lemma 2. *Let $\mathcal{X}^n$ be the space of length-n basepoint-augmented sequences in $\mathcal{X} = \mathbb{R}^d$, $\mathbf{x}, \mathbf{z} \in \mathcal{X}^n$, and $\phi : \mathcal{X} \to \mathcal{H}$ be the canonical feature map associated with kernel κ with* RKHS *$\mathcal{H}$. Assume ϕ is continuous. Then the map $\mathbf{x} \mapsto \kappa(\mathbf{x}, \cdot)$ – where $\kappa(\mathbf{x}, \cdot)$ is the linear interpolation of the points $(\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n))$ in $\mathcal{H}$ – is continuous in the one-variation topology.*

Proof. By Lemma 1, the one-variation is a norm on length-n basepoint-augmented sequences in $\mathcal{X}$. We will proceed by showing that the one-variation is an equivalent norm to the 1-product norm, defined as

$$\|\mathbf{x}\|_{\mathcal{X}^n} := \sum_{i=1}^{n} \|\mathbf{x}_i\|_{\mathcal{X}}\,, \tag{3.3}$$

which induces the product topology on $\mathcal{X}^n$. By showing this, we will have the following implications: from the definition of the 1-product norm,

$$\|\mathbf{x}-\mathbf{z}\|_{\mathcal{X}^n} < \tilde{\delta} \Longrightarrow \|\mathbf{x}_i - \mathbf{z}_i\|_{\mathcal{X}} < \tilde{\delta} \text{ also;} \tag{3.4}$$

by continuity of ϕ, we have that $\forall\, \tilde{\epsilon} > 0$, $\exists\, \tilde{\delta} > 0$ such that

$$\|\mathbf{x}_i - \mathbf{z}_i\|_{\mathcal{X}} < \tilde{\delta} \Longrightarrow \|\phi(\mathbf{x}_i) - \phi(\mathbf{z}_i)\|_{\mathcal{H}} < \tilde{\epsilon}; \tag{3.5}$$

and that choosing $\tilde{\epsilon} = \epsilon/2(n-1)$ for any $\epsilon > 0$ means that ensuring $\|\phi(\mathbf{x}_i) - \phi(\mathbf{z}_i)\|_{\mathcal{H}} < \tilde{\epsilon}$ for all i means

$$\begin{aligned}
\|\kappa(\mathbf{x},\cdot) - \kappa(\mathbf{z},\cdot)\|_{1-\text{var}} &= \sum_{i=1}^{n-1} \|(\phi(\mathbf{x}_{i+1}) - \phi(\mathbf{z}_{i+1})) - (\phi(\mathbf{x}_i) - \phi(\mathbf{z}_i))\|_{\mathcal{H}} \\
&\leq \sum_{i=1}^{n-1} \|\phi(\mathbf{x}_{i+1}) - \phi(\mathbf{z}_{i+1})\|_{\mathcal{H}} + \|\phi(\mathbf{x}_i) - \phi(\mathbf{z}_i)\|_{\mathcal{H}} \\
&< 2(n-1)\tilde{\epsilon} \\
&= \epsilon.
\end{aligned} \tag{3.6}$$

We therefore have the following chain of implications: for every $\epsilon > 0$ there is a $\tilde{\delta} > 0$ such that

$$\begin{aligned}
\|\mathbf{x}-\mathbf{z}\|_{\mathcal{X}^n} < \tilde{\delta} \Longrightarrow \|\mathbf{x}_i - \mathbf{z}_i\|_{\mathcal{X}} < \tilde{\delta} \Longrightarrow \|\phi(\mathbf{x}_i) - \phi(\mathbf{z}_i)\|_{\mathcal{H}} < \tilde{\epsilon} \\
\Longrightarrow \|\kappa(\mathbf{x},\cdot) - \kappa(\mathbf{z},\cdot)\|_{1-\text{var}} < \epsilon.
\end{aligned} \tag{3.7}$$

It therefore suffices to show that for any $\tilde{\delta} > 0$ there is a $\delta > 0$ such that $\|\mathbf{x}-\mathbf{z}\|_{1-\text{var}} < \delta \Longrightarrow \|\mathbf{x}-\mathbf{z}\|_{\mathcal{X}^n} < \tilde{\delta}$, which by this chain of implications would imply that $\forall\, \epsilon > 0$, $\exists\, \delta > 0$ such that $\|\mathbf{x}-\mathbf{z}\|_{1-\text{var}} < \delta \Longrightarrow \|\kappa(\mathbf{x},\cdot) - \kappa(\mathbf{z},\cdot)\|_{1-\text{var}} < \epsilon$. We will do so by showing that $\|\cdot\|_{1-\text{var}}$ and $\|\cdot\|_{\mathcal{X}^n}$ are equivalent norms.

We therefore seek $0 < c \leq C$ such that $c\,\|\mathbf{x}\|_{\mathcal{X}^n} \leq \|\mathbf{x}\|_{1-\text{var}} \leq C\,\|\mathbf{x}\|_{\mathcal{X}^n}$. This is trivially satisfied when $\mathbf{x} = (0, 0, \ldots, 0)$, so consider $\|\mathbf{x}\|_{\mathcal{X}^n} \neq 0$ and let $\mathbf{u} = \mathbf{x}/\,\|\mathbf{x}\|_{\mathcal{X}^n}$ such that $\|\mathbf{u}\|_{\mathcal{X}^n} = 1$. Showing that the sphere $\mathcal{S} = \{\mathbf{u} : \|\mathbf{u}\|_{\mathcal{X}^n} = 1\}$ is compact,

and that $\|\cdot\|_{1-\text{var}}$ is continuous in the product topology for $\mathcal{X}^n$, enables us to use the Extreme Value Theorem to find c and C as $\inf_{\mathbf{u}'\in\mathcal{X}^n}\|\mathbf{u}'\|_{1-\text{var}}$ and $\sup_{\mathbf{u}'\in\mathcal{X}^n}\|\mathbf{u}'\|_{1-\text{var}}$.

To show that $\mathcal{S}$ is compact, we note that $\|\mathbf{u}\|_{\mathcal{X}^n} = 1 \Rightarrow \|\mathbf{u}_i\|_{\mathcal{X}} \leq 1$. The sets $\mathcal{S}_i := \{\mathbf{u}_i : \|\mathbf{u}_i\|_{\mathcal{X}} \leq 1\}$ are closed and bounded subsets of $\mathcal{X} = \mathbb{R}^d$ and so are compact by the Heine-Borel Theorem. Then by Tychonoff's Theorem, the set $\prod_{i=1}^n \mathcal{S}_i$ is compact under the product topology (which is induced by $\|\cdot\|_{\mathcal{X}^n}$), and the sphere $\mathcal{S} \subseteq \prod_{i=1}^n \mathcal{S}_i$ is a closed subset of a compact set and is therefore also compact. Then, we show that $\|\cdot\|_{1-\text{var}}$ is continuous in the product topology by considering that for all $\epsilon > 0$, we have that $\forall \mathbf{x}, \mathbf{z} \in \mathcal{S}$

$$\|\mathbf{x}-\mathbf{z}\|_{\mathcal{X}^n} < \frac{\epsilon}{2} \Longrightarrow |\|\mathbf{x}\|_{1-\text{var}} - \|\mathbf{z}\|_{1-\text{var}}| \leq \|\mathbf{x}-\mathbf{z}\|_{1-\text{var}} \leq 2\|\mathbf{x}-\mathbf{z}\|_{\mathcal{X}^n} < \epsilon. \quad (3.8)$$

Thus, since $\|\cdot\|_{1-\text{var}}$ is a continuous function on a compact set $\mathcal{S} = \{\mathbf{u} : \|\mathbf{u}\|_{\mathcal{X}^n} = 1\}$, then by the Extreme Value Theorem it is bounded and achieves its minimum $c = \inf_{\mathbf{u}'\in\mathcal{X}^n}\|\mathbf{u}'\|_{1-\text{var}}$ and maximum $C = \sup_{\mathbf{u}'\in\mathcal{X}^n}\|\mathbf{u}'\|_{1-\text{var}}$. Thus $\forall \mathbf{x} \in \mathcal{X}^n$ with $\mathbf{u} := \mathbf{x}/\|\mathbf{x}\|_{\mathcal{X}^n} \in \mathcal{S}$,

$$c \leq \|\mathbf{u}\|_{1-\text{var}} \leq C \implies c\|\mathbf{x}\|_{\mathcal{X}^n} \leq \|\mathbf{x}\|_{1-\text{var}} \leq C\|\mathbf{x}\|_{\mathcal{X}^n} \quad (3.9)$$

and so $\|\cdot\|_{1-\text{var}}$ and $\|\cdot\|_{\mathcal{X}^n}$ are equivalent norms. In particular, we have that $\|\mathbf{x}\|_{\mathcal{X}^n} \leq \|\mathbf{x}\|_{1-\text{var}}/c$, such that for all $\tilde{\delta} > 0$, we have that

$$\|\mathbf{x}-\mathbf{z}\|_{1-\text{var}} < \delta := c\tilde{\delta} \Longrightarrow \|\mathbf{x}-\mathbf{z}\|_{\mathcal{X}^n} < \tilde{\delta}, \quad (3.10)$$

and so we are done. □

We consider next the continuity of the signature map for piecewise linear paths of bounded variation in $\mathcal{H}$. For such paths, the signature truncated at degree 1 is a multiplicative functional with bounded variation (see Lyons et al. (2002, Section 3.1.2)) and, consequently, a special case of Lyons et al. (2002, Theorem 3.1.3) applies:

Lemma 3. *Let V be a Banach space, $x, z \in BV([0,T], V)$ be two bounded variation paths in V, and τ be a constant such that*

$$\tau \geq 2\left(1 + \sum_{r=3}^{\infty}\left(\frac{2}{r-2}\right)^2\right).$$

If φ is a constant such that

$$\|x\|_{1-\text{var}}, \|z\|_{1-\text{var}} \leq \frac{\varphi}{\tau} \qquad \textit{and} \qquad \|x-z\|_{1-\text{var}} \leq \chi\frac{\varphi}{\tau}$$

for some $\chi > 0$, then for all $m \geq 1$

$$\|S_m(x) - S_m(z)\|_{V^{\otimes m}} \leq \frac{\chi}{\tau} \cdot \frac{\varphi^m}{m!}. \tag{3.11}$$

An immediate consequence of this is that the signature map is continuous in the 1-variation topology for bounded variation paths in Banach spaces:

Corollary 1. *Let $\mathcal{H}$ be a Hilbert space, $x, z \in BV([0,T], \mathcal{H})$ be two bounded variation paths in $\mathcal{H}$, and τ be as in Lemma 3. If φ is a constant such that*

$$\|x\|_{1-\text{var}},\ \|z\|_{1-\text{var}} \leq \frac{\varphi}{\tau} \qquad \textit{and} \qquad \|x - z\|_{1-\text{var}} \leq \chi \frac{\varphi}{\tau}$$

for some $\chi > 0$, then

$$\|\text{Sig}(x) - \text{Sig}(z)\| \leq \frac{\chi}{\tau} \exp\left(\frac{\varphi^2}{2}\right).$$

Proof. By definition of the norm on $\prod_{m\geq 0} \mathcal{H}^{\otimes m}$,

$$\begin{aligned}
\|\text{Sig}(x) - \text{Sig}(z)\| &= \sqrt{\sum_{m\geq 0} \|S_m(x) - S_m(z)\|^2_{\mathcal{H}^{\otimes m}}} \\
&= \sqrt{0 + \sum_{m\geq 1} \|S_m(x) - S_m(z)\|^2_{\mathcal{H}^{\otimes m}}} && (S_0(x) = 1\, \forall x \in BV([0,T], \mathcal{H})) \\
&\leq \sqrt{\sum_{m\geq 1} \frac{\chi^2}{\tau^2} \cdot \left(\frac{\varphi^m}{m!}\right)^2} && \text{(from (3.11) above)} \\
&= \frac{\chi}{\tau} \sqrt{\sum_{m\geq 1} \frac{(\varphi^2)^m}{(m!)^2}} \\
&\leq \frac{\chi}{\tau} \sqrt{\sum_{m\geq 1} \frac{(\varphi^2)^m}{m!}} && \text{(smaller denominator)} \\
&\leq \frac{\chi}{\tau} \exp\left(\frac{\varphi^2}{2}\right). && \text{(convergent series)}
\end{aligned}$$

□

We show next that the map $\rho(\text{Sig}(\mathbf{y}), \cdot) : \prod_{m\geq 0} \mathcal{H}^{\otimes m} \to \mathbb{R}_{\geq 0}$, $s \mapsto \|\text{Sig}(\mathbf{y}) - s\|^2$ is continuous. To do so, we make use of the following result:

Lemma 4. *Let κ be a uniformly bounded kernel i.e. one for which $\sup_{x\in\mathcal{X}}\sqrt{\kappa(x,x)} < \infty$, and let $\kappa(\mathbf{x},\cdot) \in \mathcal{P}([0,T],\mathcal{H})$ be a $\mathcal{H}$-valued piecewise linear path with knots at $\kappa(\mathbf{x}_i,\cdot), i = 1,\dots,n$, and $\mathrm{Sig}(\mathbf{x})$ its signature. Then*

$$\sup_{\mathbf{x}\in\mathcal{X}^n} \|\mathrm{Sig}(\mathbf{x})\| < \infty. \tag{3.12}$$

Proof. For all $\mathbf{x} \in \mathcal{X}^n$, we have

$$\begin{aligned}
\|\kappa(\mathbf{x},\cdot)\|_{1-\mathrm{var}} &= \sum_{i=1}^{n-1} \|\kappa(\mathbf{x}_{i+1},\cdot) - \kappa(\mathbf{x}_i,\cdot)\|_{\mathcal{H}} && \text{(piecewise linear)}\\
&\le \sum_{i=1}^{n-1} \|\kappa(\mathbf{x}_{i+1},\cdot)\|_{\mathcal{H}} + \|\kappa(\mathbf{x}_i,\cdot)\|_{\mathcal{H}} && \text{(triangle inequality)}\\
&= \sum_{i=1}^{n-1} \sqrt{\kappa(\mathbf{x}_{i+1},\mathbf{x}_{i+1})} + \sqrt{\kappa(\mathbf{x}_i,\mathbf{x}_i)} && \text{(reproducing property)}\\
&\le 2(n-1)\sup_{z\in\mathcal{X}} \sqrt{\kappa(z,z)}. && (\kappa \text{ bounded})
\end{aligned}$$

Let $v := 2(n-1)\sup_{z\in\mathcal{X}}\sqrt{\kappa(z,z)}$. Then $\forall \mathbf{x} \in \mathcal{X}^n$,

$$\begin{aligned}
\|\mathrm{Sig}(\mathbf{x})\| &\le \left(\sum_{m=0}^{\infty} \frac{(\|\kappa(\mathbf{x},\cdot)\|^2_{1-\mathrm{var}})^m}{(m!)^2}\right)^{\frac{1}{2}} && \text{(Proposition 3)}\\
&\le \left(\sum_{m=0}^{\infty} \frac{(v^2)^m}{m!}\right)^{\frac{1}{2}}\\
&= e^{\frac{v^2}{2}}, && \text{(exponential series)}
\end{aligned}$$

where in the first inequality we make use of the factorial decay property of signatures. We obtain the result by taking the supremum over $\mathcal{X}^n$:

$$\sup_{\mathbf{x}\in\mathcal{X}^n} \|\mathrm{Sig}(\mathbf{x})\| \le e^{\frac{v^2}{2}} < \infty.$$

□

Lemma 5. *Let κ be a uniformly bounded kernel i.e. one for which $\sup_{z\in\mathcal{X}}\sqrt{\kappa(z,z)} < \infty$, and let $\kappa(\mathbf{y},\cdot) \in \mathcal{P}([0,T],\mathcal{H})$ be the observed $\mathcal{H}$-valued piecewise linear path with $\mathrm{Sig}(\mathbf{y})$ its signature. Denote the signature kernel as*

$$k(\mathbf{x},\mathbf{z}) = \langle \mathrm{Sig}(\mathbf{x}), \mathrm{Sig}(\mathbf{z})\rangle \tag{3.13}$$

Then the distance function

$$\rho\left(\mathrm{Sig}(\mathbf{y}), \cdot\right) : \prod_{m \geq 0} \mathcal{H}^{\otimes m} \to \mathbb{R}_{\geq 0}, \quad s \mapsto \|s - \mathrm{Sig}(\mathbf{y})\|^2 \tag{3.14}$$

is Lipschitz continuous in s.

Proof.

$$\begin{aligned}
\left|\mathcal{D}(\mathbf{y}, \mathbf{x}) - \mathcal{D}(\mathbf{y}, \mathbf{z})\right| &= \left|\|\mathrm{Sig}(\mathbf{x}) - \mathrm{Sig}(\mathbf{y})\|^2 - \|\mathrm{Sig}(\mathbf{z}) - \mathrm{Sig}(\mathbf{y})\|^2\right| \\
&= \left|k(\mathbf{x}, \mathbf{x}) - k(\mathbf{z}, \mathbf{z}) + 2\left(k(\mathbf{z}, \mathbf{y}) - k(\mathbf{x}, \mathbf{y})\right)\right| \\
&\leq \left|k(\mathbf{x}, \mathbf{x}) - k(\mathbf{z}, \mathbf{z})\right| + 2\left|k(\mathbf{z}, \mathbf{y}) - k(\mathbf{x}, \mathbf{y})\right|. \quad \text{(triangle inequality)}
\end{aligned}$$

Considering the first of these terms and making use of the reproducing property and symmetry of k:

$$\begin{aligned}
\left|k(\mathbf{x}, \mathbf{x}) - k(\mathbf{z}, \mathbf{z})\right| &= \left|k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{z}) + k(\mathbf{z}, \mathbf{x}) - k(\mathbf{z}, \mathbf{z})\right| \\
&= \left|\langle k(\mathbf{x}, \cdot), k(\mathbf{x}, \cdot) - k(\mathbf{z}, \cdot)\rangle + \langle k(\mathbf{z}, \cdot), k(\mathbf{x}, \cdot) - k(\mathbf{z}, \cdot)\rangle\right| \\
&\leq \left|\langle k(\mathbf{x}, \cdot), k(\mathbf{x}, \cdot) - k(\mathbf{z}, \cdot)\rangle\right| + \left|\langle k(\mathbf{z}, \cdot), k(\mathbf{x}, \cdot) - k(\mathbf{z}, \cdot)\rangle\right| \\
&\leq \left(\|\mathrm{Sig}(\mathbf{x})\| + \|\mathrm{Sig}(\mathbf{z})\|\right) \cdot \|\mathrm{Sig}(\mathbf{x}) - \mathrm{Sig}(\mathbf{z})\|,
\end{aligned}$$

where in the penultimate and final lines we use the triangle inequality and the Cauchy-Schwarz inequality twice, respectively. Considering now the second term:

$$\begin{aligned}
\left|k(\mathbf{z}, \mathbf{y}) - k(\mathbf{x}, \mathbf{y})\right| &= \left|\langle \mathrm{Sig}(\mathbf{y}), \mathrm{Sig}(\mathbf{x}) - \mathrm{Sig}(\mathbf{z})\rangle\right| \\
&\leq \|\mathrm{Sig}(\mathbf{y})\| \, \|\mathrm{Sig}(\mathbf{x}) - \mathrm{Sig}(\mathbf{z})\|, \quad \text{(Cauchy-Schwartz)}
\end{aligned}$$

where in the first line we use the definition and symmetry of the inner product. Putting the two terms together and using Lemma 5, we have

$$\begin{aligned}
\left|\mathcal{D}(\mathbf{y}, \mathbf{x}) - \mathcal{D}(\mathbf{y}, \mathbf{z})\right| &\leq \left(\|\mathrm{Sig}(\mathbf{x})\| + \|\mathrm{Sig}(\mathbf{z})\| + 2\|\mathrm{Sig}(\mathbf{y})\|\right) \|\mathrm{Sig}(\mathbf{x}) - \mathrm{Sig}(\mathbf{z})\| \\
&\leq 4e^{\frac{v^2}{2}} \|\mathrm{Sig}(\mathbf{x}) - \mathrm{Sig}(\mathbf{z})\|
\end{aligned}$$

where v is as in Lemma 5. Thus $\rho\left(\mathrm{Sig}(\mathbf{y}), \cdot\right)$ is Lipschitz continuous. □

We finally arrive at the proof of Proposition 6:

Proof of Proposition 6. Compositions of continuous maps are continuous, and each of the constituent maps are continuous from the Lemmas and Corollaries presented above. □

Injectivity of the signature map is also guaranteed under these conditions:

Proposition 7. *Let* $\mathcal{X} := \mathbb{R}^d$, $\mathbf{x}, \mathbf{y} \in \mathcal{X}^n$. *Assume both* $\mathbf{x}$ *and* $\mathbf{y}$ *are time- and basepoint-augmented, and that* $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a continuous, uniformly bounded, injective kernel. Then* $\mathrm{Sig}(\mathbf{x}) = \mathrm{Sig}(\mathbf{y})$ *iff* $\mathbf{x} = \mathbf{y}$.

Proof. Obtaining a signature from a length-n data stream $\mathbf{x}$ entails: (1) lifting the points $\mathbf{x}_i$ in $\mathbf{x}$ to the RKHS $\mathcal{H}$ associated with κ as $\kappa(\mathbf{x}_i, \cdot)$; (2) applying a linear interpolation to obtain a piecewise linear $\mathcal{H}$-valued path $\kappa(\mathbf{x}, \cdot)$; and (3) finally taking the signature of $\kappa(\mathbf{x}, \cdot)$. To show injectivity of this composite map, it suffices to show injectivity of each of these three steps since the composition of injective maps is injective.

(1) is trivially injective, due to the assumed injectivity of κ. (2) is by definition injective for a length-n sequence in $\mathcal{H}$. To show injectivity of (3), we note that time-augmentation of the sequences, along with injectivity of κ, ensure that the lifted paths are injective, such that no tree-like equivalence is observed between the interpolated paths in $\mathcal{H}$. Time-augmentation further makes the signature sensitive to parameterisation, removing its parameterisation invariance property. Uniform boundedness of κ ensures that $\kappa(\mathbf{x}, \cdot)$ is of bounded variation, such that $\kappa(\mathbf{x}, \cdot) \in \mathcal{P}([0, T], \mathcal{H})$. To see this, note that for a piecewise linear path $\kappa(\mathbf{x}, \cdot)$,

$$\|\kappa(\mathbf{x}, \cdot)\|_{1-\mathrm{var}} = \sum_{i=1}^{n-1} \|\kappa(\mathbf{x}_{i+1}, \cdot) - \kappa(\mathbf{x}_i, \cdot)\|_{\mathcal{H}} \leq 2(n-1) \sup_{\mathbf{z} \in \mathcal{X}} \sqrt{\kappa(\mathbf{z}, \mathbf{z})} < \infty,$$

where we have used the reproducing property of κ and the triangle inequality. Finally, since basepoint augmentation makes the signature sensitive to paths that differ only by translations, the desired result follows from Theorem 3. □

Taken together, these results provide the same guarantees for the asymptotic correctness of the S-ABC posterior as $\varepsilon \to 0$ for dynamic, stochastic simulators as, for example, the Wasserstein ABC posterior of Bernton et al. (2019).

3.1.1.2 Behaviour as $n \to \infty$ for fixed ε

We now consider the behaviour of the S-ABC posterior as the rate at which a (continuous) path is sampled tends to infinity, such that $n \to \infty$ within a fixed, finite time interval $[0, T]$. For the moment, we will assume that the continuous $\mathcal{H}$-valued paths h, g of which $\kappa(\mathbf{x}, \cdot)$ and $\kappa(\mathbf{z}, \cdot)$ are discretisations are of bounded variation, and will discuss a more general setting later. Throughout this section, we will denote with $\zeta(0, T)$ a partition of the interval $[0, T]$, $\Delta_{[0,T]} := \{(s, t) \in [0, T]^2 : 0 \leq s \leq t \leq T\}$, $\text{mesh}(0, T)$ the largest interval in $\zeta(0, T)$ i.e.

$$\text{mesh}(0, T) := \max_{(s,t)\in\zeta(0,T)} |t - s|$$

with $0 \leq s \leq t \leq T$,

$$\mathcal{D}(\mathbf{x}, \mathbf{y}) = \|\text{Sig}(\mathbf{x}) - \text{Sig}(\mathbf{y})\|^2, \qquad \mathbf{x}, \mathbf{y} \in \mathcal{P}([0, T], \mathcal{X})$$

and

$$\mathcal{D}(h, g) = \|\text{Sig}(h) - \text{Sig}(g)\|^2, \qquad h, g \in BV([0, T], \mathcal{H}).$$

Lemma 6. *Let κ be a uniformly bounded, injective kernel on $\mathcal{X}$, and $\kappa(\mathbf{x}, \cdot), \kappa(\mathbf{y}, \cdot) \in \mathcal{P}([0, T], \mathcal{H})$ be the simulated and observed datasets, respectively, which are discretisations of underlying paths $h, g \in BV([0, T], \mathcal{H})$. Then,*

$$\|\text{Sig}(\mathbf{x}) - \text{Sig}(\mathbf{y})\|^2 \longrightarrow \|\text{Sig}(h) - \text{Sig}(g)\|^2 \tag{3.15}$$

as $\text{mesh}(0, T) \to 0$.

Proof. Let $\rho = (s_i)_{i=1}^N, 0 = s_1 < \cdots < s_N = T$ and $\varrho = (t_j)_{j=1}^M, 0 = t_1 < \cdots < t_M = T$ be partitions of the interval $[0, T]$ such that $\kappa(\mathbf{x}, \cdot)_{s_i} = h_{s_i}, i = 1, \ldots, N$ and $\kappa(\mathbf{y}, \cdot)_{t_j} = g_{t_j}, j = 1, \ldots, M$, with the paths $\kappa(\mathbf{x}, \cdot), \kappa(\mathbf{y}, \cdot) \in \mathcal{P}([0, T], \mathcal{H})$ linear in between these points. Then, by Kiŕaly and Oberhauser (2019, Corollary 4.7),

$$|k(\mathbf{x}, \mathbf{y}) - k(h, g)| \leq 2e^{\|h\|_{1-\text{var}}+\|g\|_{1-\text{var}}} - e^{\|h\|_{1-\text{var}}+\|\kappa(\mathbf{y},\cdot)\|_{1-\text{var}}} - e^{\|\kappa(\mathbf{x},\cdot)\|_{1-\text{var}}+\|g\|_{1-\text{var}}},$$

where convergence is uniform and of order $O(\max_i \|h_{[s_i,s_{i+1}]}\|_{1-\text{var}}+\max_j \|g_{[t_j,t_{j+1}]}\|_{1-\text{var}})$. Therefore,

$$\begin{aligned}
\big|\mathcal{D}(\mathbf{x}, \mathbf{y}) - \mathcal{D}(h, g)\big| &= \big|k(\mathbf{x}, \mathbf{x}) - k(h, h) + k(\mathbf{y}, \mathbf{y}) - k(g, g) + 2(k(h, g) - k(\mathbf{x}, \mathbf{y}))\big| \\
&\leq \big|k(\mathbf{x}, \mathbf{x}) - k(h, h)\big| + \big|k(\mathbf{y}, \mathbf{y}) - k(g, g)\big| + 2\big|k(h, g) - k(\mathbf{x}, \mathbf{y})\big| \\
&\longrightarrow 0 \quad \text{as} \quad \text{mesh}(0, T) \longrightarrow 0,
\end{aligned}$$

where the triangle inequality is used in the second line. □

As a consequence of the above, we see that the S-ABC posterior for piecewise linear paths converges to the S-ABC posterior for continuous paths of bounded variation as the sampling rate is increased indefinitely:

Proposition 8. *Let κ be a uniformly bounded, injective kernel and $g \in BV([0,T], \mathcal{H})$ be the limit of $\kappa(\mathbf{y}, \cdot) \in \mathcal{P}([0,T], \mathcal{H})$ as* $\text{mesh}(0,T) \to 0$. *Then for fixed $\varepsilon > 0$ such that*

$$\varepsilon > \inf_{h' \in BV([0,T],\mathcal{H})} \mathcal{D}(h', g)$$

and as $n \to \infty$ ($\text{mesh}(0,T) \to 0$*), the* S-ABC *posterior*

$$\pi(\boldsymbol{\theta} \mid \mathcal{D}(\mathbf{x}, \mathbf{y}) \leq \varepsilon) \rightharpoonup \pi(\boldsymbol{\theta} \mid \mathcal{D}(h, g) \leq \varepsilon)$$

for $h, g \in BV([0,T], \mathcal{H})$, where $\rightharpoonup$ denotes weak convergence.

Proof. By Lemma 6, $\mathcal{D}(\mathbf{x}, \mathbf{y}) \to \mathcal{D}(h, g)$ where $h, g \in BV([0,T], \mathcal{H})$ are the bounded variation paths of which $\kappa(\mathbf{x}, \cdot)$ and $\kappa(\mathbf{y}, \cdot)$ are discretisations. Further, by our choice of ε, $\mathbb{P}(\mathcal{D}(h, g) = \varepsilon) = 0$ and $\mathbb{P}(\mathcal{D}(h, g) < \varepsilon) > 0$, where $\mathbb{P}$ denotes a probability measure. Then, we follow Miller and Dunson (2018) and apply Lemma 5.1 contained therein using the same notation: we obtain the result by taking the $(U_n)_{n \geq 1}$ to be the $\mathcal{D}(\mathbf{x}, \mathbf{y})$ as $\text{mesh}(0,T)$ decreases and $\|\kappa(\mathbf{x}, \cdot)\|_{1-\text{var}}, \|\kappa(\mathbf{y}, \cdot)\|_{1-\text{var}} \to \|h\|_{1-\text{var}}, \|g\|_{1-\text{var}}$; $U = \mathcal{D}(h, g)$; $V = \varepsilon$; and $W = h(\boldsymbol{\theta})$ for any continuous, bounded $h : \boldsymbol{\Theta} \to \mathbb{R}$. □

This result shows that for fixed ε greater than the minimum possible value for $\mathcal{D}(h, g)$, the S-ABC posterior does not converge to a Dirac mass in the limit of infinite data over a fixed finite time horizon, or as the sampling rate is increased indefinitely in the interval $[0, T]$. Furthermore, by the same reasoning as in Miller and Dunson (2018, Theorem 5.6), continuity of the signature in the 1-variation topology (see Appendix 6 and Lyons et al. (2002, Section 3.1.2)) implies that the S-ABC posterior is robust to small changes in the data even in the limit of infinite data. As the authors discuss, this can be advantageous in misspecified settings, which is typically the case in real-world modelling and inference problems.

Remark 4. *Throughout the above, we have assumed that the limiting paths are of bounded variation as* $\text{mesh}(0,T) \to 0$. *We may consider a more general case by adopting the weaker assumption that the limiting paths h and g for $\kappa(\mathbf{x}, \cdot)$ and $\kappa(\mathbf{y}, \cdot)$ as* $\text{mesh}(0,T) \to 0$ *are* geometric *p-rough paths (see Appendix A). By the Extension*

Theorem (see Appendix A and Lyons et al. (2002, Theorem 3.1.3)), the iterated integrals comprising the geometric p-rough paths h and g may be extended to all iterated integrals to obtain a path signature for h and g that is continuous in the p-variation topology (see Lyons et al. (2007, Theorem 3.10) and Appendix A). In this way, we may obtain S-ABC *posteriors in the limit* $\text{mesh}(0,T) \to 0$ *for classes of models that are much "rougher" than the bounded variation case considered so far, such as continuous semimartingales, Gaussian processes, continuous-time Markov processes etc. The "coarsened" posteriors (using the nomenclature introduced by Miller and Dunson (2018)) resulting from the application of* S-ABC *in these instances are equipped with the same continuity property, now in the p-variation topology, that the* S-ABC *posterior enjoyed in the bounded variation case under the 1-variation topology.*

3.1.2 Signature Regression ABC

In some circumstances, it is desirable to find low-dimensional summary statistics for use in ABC. For example, Fearnhead and Prangle (2012) propose the use of the posterior mean $\mathbb{E}[\boldsymbol{\theta} \mid \mathbf{y}]$ as a summary statistic for $\mathbf{y}$, since it is an optimal choice in that it minimises the quadratic loss between the ABC posterior mean and the true parameter. As discussed in Section 1.5.1, this involves fitting a vector-valued regression model from a large candidate set of summary statistics to parameters $\boldsymbol{\theta}$, since this generates an estimate of the (unknown) posterior mean. The approach of Fearnhead and Prangle (2012) belongs to a larger class of methods for generating low-dimensional summary statistics from a large initial candidate set, sometimes termed "projection methods" (Beaumont, 2019), which also includes the partial least regression method proposed by Wegmann et al. (2009).

However, a significant problem with projection methods is that it is often unclear which summary statistics should be included in the initial candidate set. Yet, the efficacy of the approach requires this initial candidate set to contain informative summaries in the first place. Contriving informative statistics thus represents a major obstacle in many inference tasks, and can involve significant domain expertise, experimentation, and computational expense. Consequently, when low-dimensional summary statistics are desired, it would be preferable to bypass the manual construction of an initial candidate set of statistics in order to use projection methods.

For the case of time series models, the path signature is a natural set of summary statistics for the regression task in SA-ABC, providing a basis for learning functions on streams due to its unique universal nonlinearity property. Naive regression on the full path signature is of course impossible, since the signature is an infinite-dimensional object. However, this may once again be circumvented using the signature kernel and corresponding kernel trick (see Definition 3), in the following way: use the signature kernel and kernel ridge regression (Hastie et al., 2001) to implicitly regress parameters onto the *full* signature, which is in a sense equivalent to using the infinitely long path signature as the candidate set of summary statistics in semi-automatic ABC. That is, using training examples $\{\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}\}_{i=1}^{R} \sim p(\mathbf{x} \mid \boldsymbol{\theta})\,\pi(\boldsymbol{\theta})$, we find a function $\hat{\boldsymbol{\theta}}_j$ in the RKHS associated with the signature kernel k, which by the Representer Theorem has the following form for each component $\boldsymbol{\theta}_j, j = 1, \ldots, p$ of the p-dimensional parameters $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{R}$:

$$\hat{\boldsymbol{\theta}}_j(\mathbf{x}) = \sum_{i=1}^{R} \boldsymbol{\omega}_i^{(j)} k(\mathbf{x}, \mathbf{x}^{(i)}) \tag{3.16}$$

with

$$\boldsymbol{\omega}^{(j)} = (\mathbf{G} + \alpha \mathbf{I}_R)^{-1} \boldsymbol{\psi}^{(j)}, \qquad \mathbf{G}_{mn} = k(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}), \tag{3.17}$$

$$\boldsymbol{\psi}^{(j)} = \begin{bmatrix} \boldsymbol{\theta}_j^{(1)} \\ \boldsymbol{\theta}_j^{(2)} \\ \vdots \\ \boldsymbol{\theta}_j^{(R)} \end{bmatrix}, \qquad \mathbf{I}_R = \operatorname{diag}(1, 1, \ldots, 1) \in \mathbb{R}^{R \times R}, \tag{3.18}$$

and $\alpha \geq 0$ is a regularisation parameter to be tuned. In this sense, signatures not only provide a natural notion of distance between time series, as described in Section 3.1.1, but additionally provide a suitable basis for learning functions on sequences, enabling the semi-automatic construction of summary statistics. This approach to ABC is somewhat similar to that of Nakagome et al. (2013), who employ kernel ridge regression with a Gaussian RBF kernel to perform SA-ABC. Our approach differs substantially, however, in that Nakagome et al. (2013) propose the use of hand-crafted summary statistics as input to the kernel ridge regression model, while we propose the use of the full data.

In more detail, we proceed as follows:

1. fit a kernel ridge regression model using training data $\{\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}\}_{i=1}^{R} \sim p(\mathbf{x} \mid \boldsymbol{\theta})\,\pi(\boldsymbol{\theta})$. This amounts to solving the following optimisation problem for each of the p

components $j = 1, \ldots, p$ of the $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{R}$:

$$\min_{\hat{\boldsymbol{\theta}}_j \in \mathcal{H}_k} \sum_{i=1}^{R} \left(\boldsymbol{\theta}_j^{(i)} - \hat{\boldsymbol{\theta}}_j \left(\mathbf{x}^{(i)} \right) \right)^2 + \alpha \| \hat{\boldsymbol{\theta}}_j \|_{\mathcal{H}_k}^2, \tag{3.19}$$

where k is the signature kernel, $\mathcal{H}_k$ is the RKHS associated with k, and $\hat{\boldsymbol{\theta}}_j$ is the solution given by Equations (3.16)-(3.18);

2. summarise the observation $\mathbf{y}$ and all future simulations $\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$ using this trained kernel ridge regression model, i.e. use

$$\mathbf{s}(\mathbf{x}) = \begin{bmatrix} \hat{\boldsymbol{\theta}}_1(\mathbf{x}) \\ \hat{\boldsymbol{\theta}}_2(\mathbf{x}) \\ \vdots \\ \hat{\boldsymbol{\theta}}_p(\mathbf{x}) \end{bmatrix}; \tag{3.20}$$

3. use the squared difference between the summaries of $\mathbf{y}$ and $\mathbf{x}$ as the measure of discrepancy between simulation and observation,

$$\rho(\mathbf{s}(\mathbf{y}), \mathbf{s}(\mathbf{x})) = \| \mathbf{s}(\mathbf{y}) - \mathbf{s}(\mathbf{x}) \|_2^2. \tag{3.21}$$

Once the data is summarised using this regression model, the discrepancy between simulation and observation is then computed as the Euclidean distance between their corresponding outputs from the kernel ridge regression model. We herein refer to this approach as signature regression ABC (SR-ABC).

3.1.3 Computational complexity

Evaluating the signature kernel for two streams $\mathbf{y} \in \mathcal{X}^n$ and $\mathbf{x} \in \mathcal{X}^m$ with $\mathcal{X} = \mathbb{R}^d$ has complexity that is linear in d and linear in the product nm (Salvi et al., 2021). This is likewise the case for MMD, which has complexity $\mathcal{O}(n^2)$ (Park et al., 2016), and compares favourably with Wasserstein distance (WASS), which in multivariate settings is known to scale poorly with the number of data. Bernton et al. (2019), for example, note costs of order n^3 when the Hungarian algorithm is used to solve the assignment problem. Alternative algorithms with favourable performance (compared to the Hungarian algorithm) are an active area of research, however scalability with data remains a problem for the application of Wasserstein ABC in large data settings.

3.2 Experiments

In this section, we present experiments comparing the performance of our signature-based methods against alternative notions of distance between simulation and observation. In particular, we compare our methods, signature ABC (S-ABC) and signature regression ABC (SR-ABC), against the use of WASS (Bernton et al., 2019) and MMD (Park et al., 2016) as measures of discrepancy, along with SA-ABC (Fearnhead and Prangle, 2012).

3.2.1 Implementation details

For all signature kernel computations, we use the `sigkernel` package (Salvi et al., 2021) and we normalise the time series by dividing by the range of the simulation output when this is known or, when this is unknown, with the expected range of the training set of size $R = 300$ for SR-ABC or $R = 300$ samples from the prior predictive distribution for S-ABC.

In order to remove the translation invariance and time-invariance properties of the signature, discussed in Section 2.2.2.2, we apply basepoint and time-augmentations to all time series in every experiment.

Unless stated otherwise, we take κ to be a Gaussian RBF kernel with scale hyperparameter σ. To tune σ and the regularisation hyperparameter for SR-ABC, we perform a grid search with 5-fold cross-validation on the training set. For S-ABC, we use the median of all pairwise Euclidean distances between points in the observation $\mathbf{y}$ for σ, although we note that other approaches could be taken, such as using the same method as for SR-ABC.

Both SA-ABC and SR-ABC require training data; for both we use $R = 300$ training examples $\{\mathbf{x}^{(j)}, \boldsymbol{\theta}^{(j)}\}_{j=1}^{R} \sim p(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$. When $\pi(\cdot)$ has bounded support, we normalise the parameters $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{R}$ in the training set with the range of the prior in each dimension. We also tune the bandwidth parameter for the Gaussian RBF kernel employed in the MMD loss for K2-ABC using the median of the pairwise absolute differences between observations in $\mathbf{y}$, as recommended by Park et al. (2016).

In all experiments, WASS indicates the 1-Wasserstein distance with curve matching,

Algorithm 4: Rejection sampling scheme

Input: prior π, observation $\mathbf{y}$, distance function $\mathcal{D}(\cdot, \cdot)$, number of particles N, final sample size $M < N$;
Result: Empirical posterior $\sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^{(i)}}$
for $i = 1, \ldots, N$ **do**
  Sample $\boldsymbol{\theta}^{(i)} \sim \pi(\boldsymbol{\theta})$;
  Simulate $\mathbf{x}^{(i)} \sim p(\mathbf{x} \mid \boldsymbol{\theta}^{(i)})$;
  Evaluate distance $\mathcal{D}(\mathbf{x}^{(i)}, \mathbf{y})$;
end
Retain the M particles $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{M}$ with the lowest losses

which as described in Section 1.5.1 is a method for using the Wasserstein distance for time series recommended in Bernton et al. (2019). To determine the λ coefficient, we follow the guidance of Thorpe et al. (2017) and choose

$$\lambda \simeq \frac{V}{T}, \tag{3.22}$$

where V is the expected vertical range and T is the length of the time interval over which observations are made, in order to balance the effects of vertical and horizontal transport. Where the value of V is not apparent *a priori*, we estimate it using $R = 300$ samples from the prior predictive distribution. Distances are computed using the Python Optimal Transport package (Flamary et al., 2021).

For all losses, we sample from the ABC posterior using the simple rejection scheme outlined in Algorithm 4 and, unless stated otherwise, use $N = 10^5$ and $M = 10^3$. While other, more sophisticated schemes exist, we choose this to facilitate a simple and transparent comparison of the different distance measures. To assess the quality of the recovered posteriors, we compute the 1-Wasserstein distance and an unbiased estimate of the maximum mean discrepancy (MMD) between the approximate ground truth posteriors $\hat{\pi}_{\cdot|\mathbf{y}}$ and empirical posteriors $\hat{\pi}_{\text{ABC}}$. In both cases, smaller values indicate a closer match to the approximate ground truth. To estimate the MMD between posteriors, we use a Gaussian RBF kernel with scale parameter chosen according to the median heuristic (Briol et al., 2019).

3.2.1.1 Reference posteriors using MCMC

We provide details on the schemes employed to generate approximate ground-truth posterior densities for some of the experiments we present below.

Metropolis-Hastings For the GBM and Brock & Hommes models, we obtain samples from the ground truth posterior using MH. We follow the guidelines of Schmon and Gagnon (2021) and use a multivariate normal proposal, for which we estimate the covariance matrix using a pilot run. We subsequently tune the MH algorithm according to Schmon and Gagnon (2021, Table 1) and run the MH for 10^5 steps, keeping a thinned subset of 10^3 samples as our baseline.

3.2.2 Ricker model

As the first example, we use the Ricker model, parameter prior, and simulated dataset as described in Section 1.7.1.

The time series generated by the Ricker model tend to consist of many zero terms, with occasional spikes. For this reason, we use the cumulative sum pre-signature transformation (see Section 2.2.4.1) for S-ABC, which is a common transformation for spiking data such as medical data (Morrill et al., 2019). In our experiments, we also found that the WASS- and MMD-based methods benefitted from this transform and were not competitive without it. We therefore also report the results obtained with WASS and MMD with this cumulative sum transform applied. For SA-ABC, the hand-crafted summary statistics we use are those proposed in Wood (2010), and consist of: the autocovariances to lag 5; the mean; the number of zeros in the sequence; the coefficients of the regression $\mathbf{x}_{t+1}^{0.3} = \beta_1 \mathbf{x}_t^{0.3} + \beta_2 \mathbf{x}_t^{0.6} + \epsilon_t$ for error term ϵ_t; and the coefficients of the cubic regression of the ordered differences $\mathbf{x}_t - \mathbf{x}_{t-1}$ on their observed values.

In Figure 3.1, we show boxplots for the Wasserstein distances and MMDs between samples from the ABC posteriors – denoted with $\hat{\pi}_{\text{ABC}}$ – and samples from an approximation of the true posterior obtained using PMCMC (Andrieu et al., 2010, see Section 3.2.1.1 for details), which we denote with $\hat{\pi}_{\cdot|\mathbf{y}}$. We also show boxplots for the Euclidean distances between the ABC posterior means and the PMCMC posterior mean. These boxplots are all obtained by running the ABC procedure 20 times with different seeds for each distance measure.

From this, we see that the signature-based methods tend to produce better performance across all three metrics considered. In more detail, the estimate of the approximate ground truth posterior obtained with the signature-based methods are

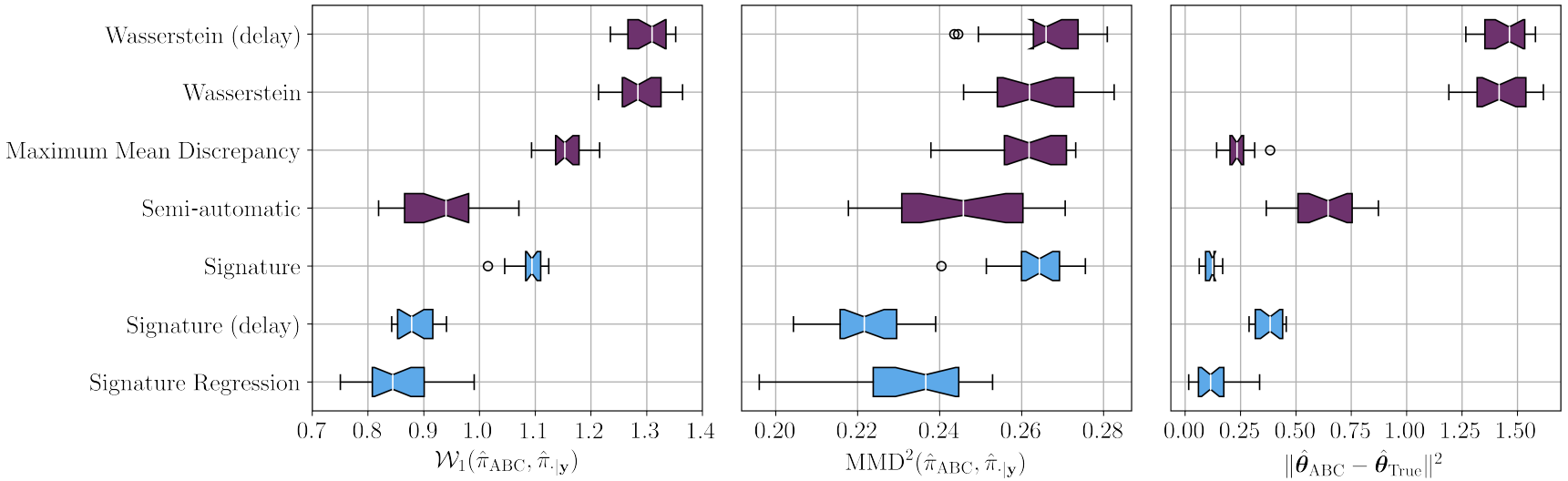


Figure 3.1: (Ricker model) Left: Wasserstein distances between the posteriors recovered from the different distance measures and an approximate ground truth obtained using PMCMC. **Middle**: Maximum mean discrepancies between the posteriors recovered from the different distance measures and an approximate ground truth obtained using PMCMC. **Right**: Squared distances between the means of the ABC posteriors and the posterior mean obtained using a PMCMC. Our methods are shown in blue.

more accurate than MMD and WASS, as reflected in the Wasserstein distances and MMDs. For S-ABC, this performance gap is enhanced with the additional application of a lag-1 delay transformation (indicated with suffix "(delay)" in Figure 3.1 and subsequent Figures) while no such improvement is observed when applied to WASS. We note that SA-ABC performs particularly well in this example, as a consequence of its use of hand-crafted summary statistics developed specifically for this simulation model. However, the potential power of our signature-based methods is demonstrated by the fact that SR-ABC is able to outperform SA-ABC in all three metrics, despite the latter using summary statistics carefully engineered by experts. Finally, we observe more accurate estimates of the true posterior mean using our signature-based methods than using WASS and SA-ABC, despite the latter using summary statistics carefully engineered by experts to provide accurate inferences for this model. The posterior mean estimates from S-ABC without the delay transformation and SR-ABC are also more accurate than those of MMD, further evidencing the usefulness of our signature-based methods.

3.2.3 Geometric Brownian motion

Here, we consider the geometric Brownian motion (GBM) model and inference task as described in Section 1.7.2.1.

For SA-ABC, we follow Fearnhead and Prangle (2012) and regress the parameters $\boldsymbol{\theta}$ onto the first, second, third, and fourth powers of summary statistics of the time series. Specifically, we take the first, second, third, and fourth powers of the variance and lag-1 and -2 autocorrelations of the increments of the log time series, $\log\left(\mathbf{x}_{i\Delta t}/\mathbf{x}_{(i-1)\Delta t}\right)$, since these are informative of the parameters being inferred.

We show in Figure 3.2 the marginal posteriors recovered using the Metropolis-Hastings (MH) approximation (see Section 3.2.1.1 for details) and the true likelihood function, along with the approximate posteriors obtained using the rejection sampling scheme in Algorithm 4 and each of the distance measures considered. The suffix "(delay)" once again indicates that the lag-1 delay transformation was applied. From this, we see that and SR-ABC and S-ABC track the shape of the approximate ground truth marginal posterior generated by MH for μ more closely than all other methods, and that the marginal distribution for σ concentrates in the neighbourhood of the approx-

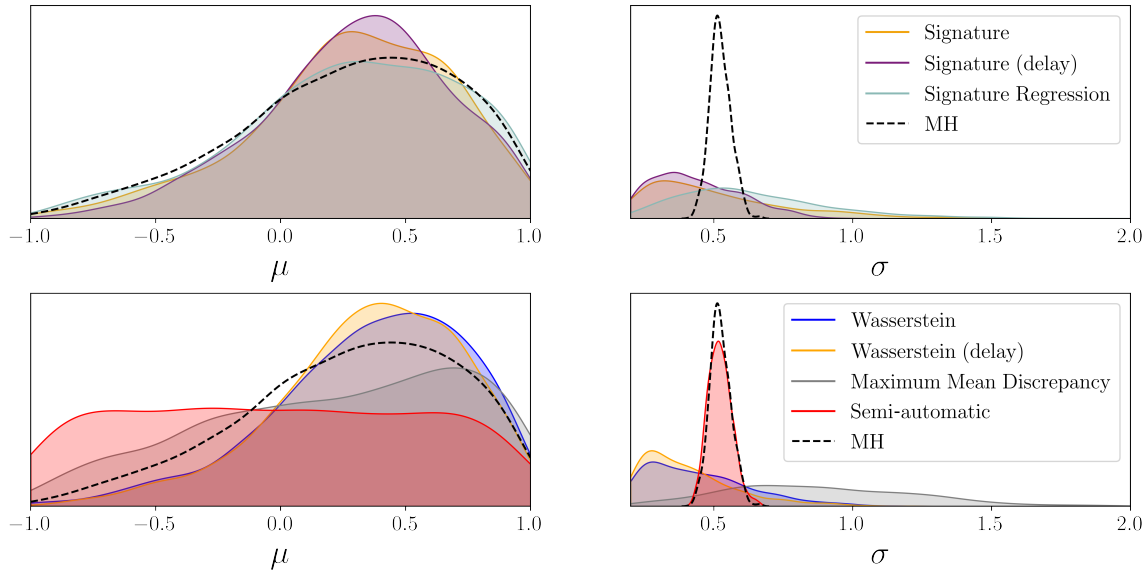


Figure 3.2: (Geometric Brownian motion) Examples of the marginal posterior distributions recovered using each loss function and the approximate ground-truth posterior recovered with a Metropolis-Hastings (MH) random walk. Top: The marginal posteriors recovered using our signature methods (S-ABC and SR-ABC) and the approximate ground-truth posterior (MH). Bottom: The marginal posteriors recovered using the Wasserstein distance with curve matching (WASS), K2-ABC (MMD), and semi-automatic ABC with powers of the variance and lag-1 and -2 autocorrelations of the increments of the log time series as regressors (SA-ABC).

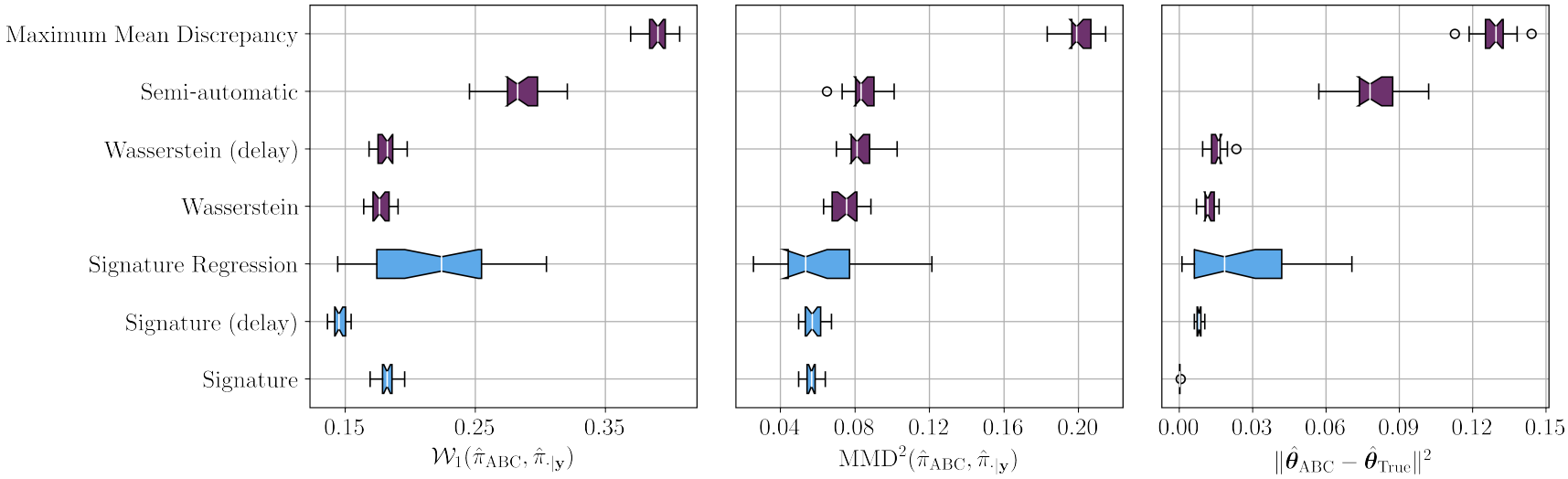


Figure 3.3: (Geometric Brownian motion) Left: Wasserstein distances between the posteriors recovered from the different distance measures and an approximate ground truth obtained using MH. **Middle**: Maximum mean discrepancies between the posteriors recovered from the different distance measures and an approximate ground truth obtained using MH. **Right**: Squared distances between the means of the ABC posteriors and the posterior mean obtained using MH. Our methods are shown in blue.

imate ground-truth marginal posterior for σ. This is in contrast to, for example, the MMD, which is overly dispersed and biased for σ.

In this example, SA-ABC has been able to very accurately approximate the marginal density for σ as a consequence of the informative set of summary statistics provided to this method. However, SA-ABC has experienced difficulty recovering the shape of the marginal density for μ, despite the provided summary statistics also being informative of this parameter. The fact that the signature- and Wasserstein-based methods are able to outperform SA-ABC, despite the advantage the latter has been afforded, illustrates the potential power of these methods in cases where the model structure is too complex to easily derive summary statistics that are informative of the parameters.

In Figure 3.3, we show boxplots for the Wasserstein distances and MMDs between the different ABC posteriors and the approximate ground truth posterior obtained with MH, in addition to the Euclidean distance between the ABC posterior means and the MH posterior mean. The boxplots were generated by repeating the REJ-ABC procedure for each distance measure with 20 different random seeds. We see that the superior shape of the signature-based distances also manifests as lower Wasserstein distances and MMDs between their corresponding ABC posteriors and the MH posterior. Indeed, we see that S-ABC with the lag-1 delay transformation uniformly dominates the non-signature methods across all three metrics.

3.2.4 The Brock & Hommes agent-based model

In this experiment, we consider the Brock & Hommes model described in Section 1.7.3 with $\beta = 10$. We consider the task of estimating the posterior $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, where $\boldsymbol{\theta} = (g_2, b_2, g_3, b_3)$, $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n) \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*)$ is the pseudo-observation, $n = 100$, and $\boldsymbol{\theta}^* := (-0.7, -0.4, 0.5, 0.3)$ is the parameter setting used to generate $\mathbf{y}$.

We show in Figure 3.4 boxplots for the Wasserstein distance and MMD between the ABC posteriors, denoted with $\hat{\pi}_{\text{ABC}}$, and the approximate ground-truth posterior obtained with MH, denoted with $\hat{\pi}_{\cdot|\mathbf{y}}$. We also show boxplots for the Euclidean distance between the ABC posterior means and the MH posterior mean. These boxplots were created by running the REJ-ABC algorithm with the same 20 random seeds. In this

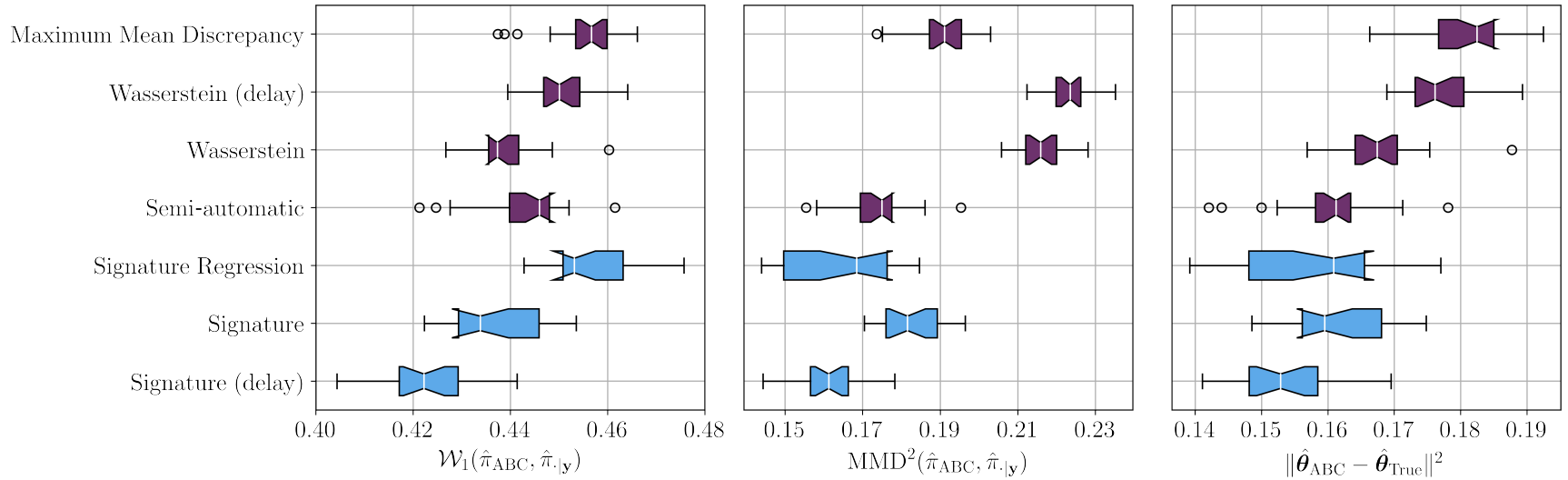


Figure 3.4: (Brock & Hommes) Left: Wasserstein distances between the posteriors recovered from the different distance measures and samples from the exact posterior. **Middle**: Maximum mean discrepancies between the posteriors recovered from the different distance measures and samples from the exact posterior. **Right**: Squared distances between the means of the ABC posteriors and the exact posterior mean. Our methods are shown in blue.

experiment, SA-ABC uses the first and second powers of l evenly spaced order statistics of the output data $\mathbf{x}$, as considered in Fearnhead and Prangle (2012), where we take $l = 10$.

From this, we see that the signature-based methods tend to generate lower values in all three metrics compared to existing methods. In particular, we see that S-ABC with the lag-1 delay transformation once again dominates existing methods uniformly across all three metrics, while the same transformation applied to WASS does not result in the same improvement. This demonstrates the potential power of our signature-based methods as automatic distance measures for ABC for dynamic, stochastic simulators.

3.2.5 An example of irregular, multivariate data: generalised stochastic epidemics

As previously discussed, the signature method naturally allows for inference with multivariate and/or irregularly spaced time series. To demonstrate this, we consider the generalised stochastic epidemic model and inference task as described in Section 1.7.4, since this is a simulator generating multivariate sequences with a random number of irregularly spaced observations.

To perform S-ABC, we bring all three channels of the multivariate stream — the number of infected individuals, number of recovered individuals, and time — into the range $[0, 1]$ by dividing by Z, Z, and T, respectively. For WASS, we set $\lambda = 2$, since the expected vertical range is approximately twice that of the horizontal range $T = 50$ when $Z = 100$.

We show in Figure 3.5 boxplots for the Wasserstein distances and MMDs between samples from WASS and S-ABC posteriors and samples from the exact posterior. We also show boxplots for the distribution of squared distances between the posterior means obtained with WASS and S-ABC and the exact posterior mean. (In this experiment, we observed the ABC posterior obtained with the MMD distance measure to perform considerably worse than WASS and S-ABC, and therefore omit these results from Figure 3.5 for clarity.) To obtain these approximate posteriors, we run Algorithm 4 with $N = 10^5$ and $M = 100$ for 20 different seeds. We also show contour plots obtained by running the inference procedure at these 20 different seeds and

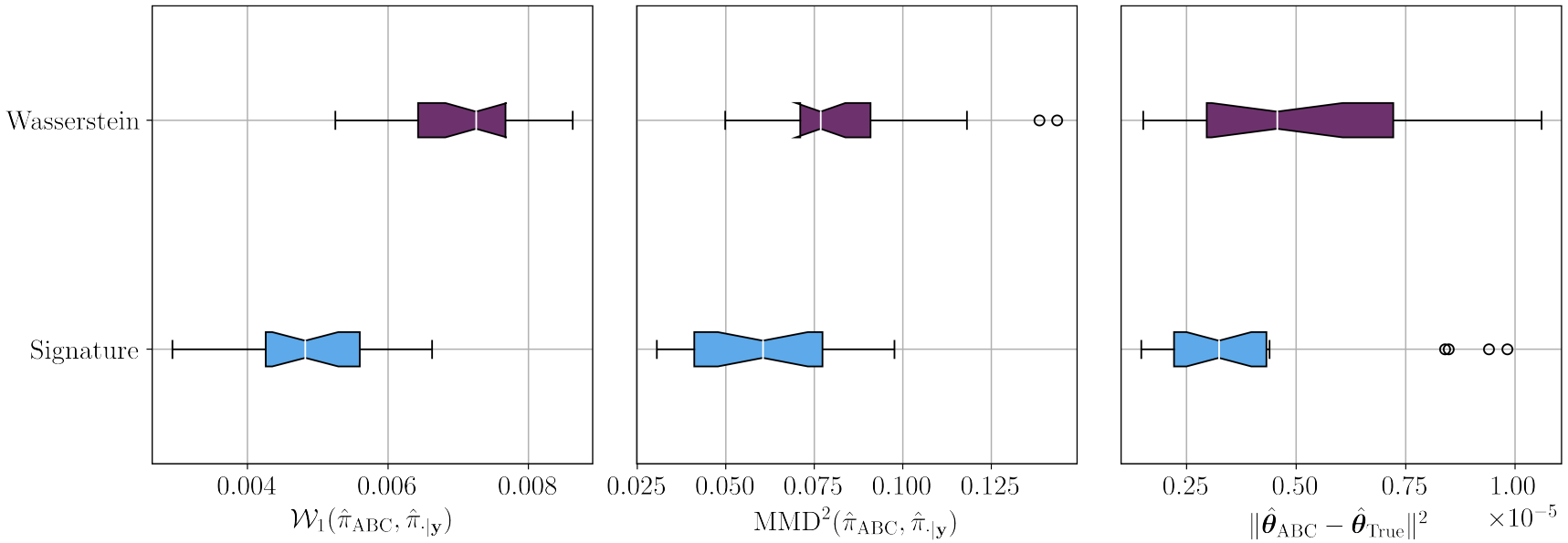


Figure 3.5: (Generalised stochastic epidemic model) Left: Wasserstein distances between the posteriors recovered from the different distance measures and samples from the exact posterior. **Middle**: Maximum mean discrepancies between the posteriors recovered from the different distance measures and samples from the exact posterior. **Right**: Squared distances between the means of the ABC posteriors and the exact posterior mean. Our method is shown in blue.

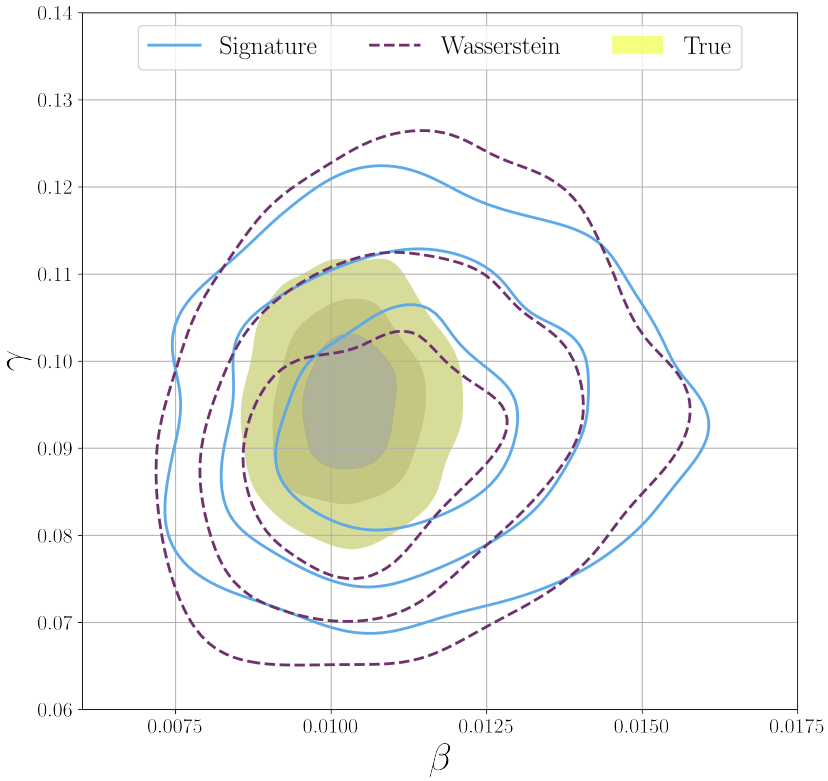


Figure 3.6: (Generalised stochastic epidemic model) The joint posterior densities recovered with the Wasserstein distance (dashed purple lines) and Signature ABC (solid blue lines), and samples from the exact posterior (filled yellow contours).

pooling the best M losses from each in Figure 3.6, along with samples from the exact posterior, (1.72).

From all of this, we see that the natural notion of distance between multivariate and irregularly sampled time series data of different lengths, enabled by the use of path signatures, manifests as better recovery of both the true posterior distribution and the true posterior mean in this example, in which the Wasserstein distances and MMDs between posteriors and Euclidean distances between posterior means for S-ABC are generally lower than those obtained using WASS.

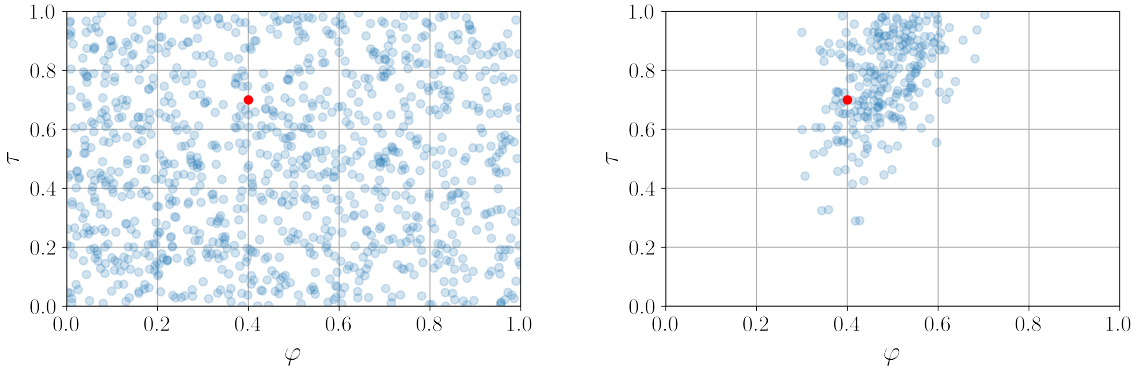


Figure 3.7: (Dynamic graph model) Samples from the prior (left) and the posterior obtained from S-ABC (right).

3.2.6 A dynamic graph model

In the previous experiments, we have seen that our signature-based methods are able to outperform existing approaches to ABC for time series simulators that generate sequential data of various types, ranging from simpler cases such as regularly spaced, univariate sequences of fixed length to more complex sequential data such as irregularly spaced, multivariate sequences of random length. However, a further consequence and benefit of the kernelisation of our signature-based approaches is that such methods can be applied to more exotic problems, in which the data evolves in more general topological spaces. For example, equipped with a suitable kernel on graphs, we may apply our signature-based methods to parameter inference problems that arise for *dynamic graph simulators* that have intractable likelihood functions.

As an illustration of this point, we take as a final example a simple dynamic graph model described in Zhang et al. (2017), which can be seen as the dynamic counterpart to the canonical Erdős-Rényi random graph model (Erdős and Rényi, 1959, 1960). In this model, edges appear with probability φ at time $t = 1, \ldots, n$ where they were absent at time $t-1$, or remain absent with probability $1-\varphi$. Similarly, edges that were present at time $t-1$ disappear with probability τ at time t or remain present with probability $1-\tau$. The output of the simulator can thus be taken as, for example, the sequence of graph snapshots or, equivalently, their adjacency matrices $\mathbf{A}_t$ in which $[\mathbf{A}_t]_{ij}$ = the number of times edge (i, j) has appeared across all time steps $t' = 0, \ldots, t$, where $\mathbf{A}_0$ is some initial seed network.

We consider the task of estimating the posterior $\pi(\boldsymbol{\theta} \mid \mathbf{A})$ for parameters $\boldsymbol{\theta} := (\varphi, \tau)$

given some observation $\mathbf{A} := (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_n) \sim p(\mathbf{B} \mid \boldsymbol{\theta}^*)$, where $n = 25$, $\mathbf{A}_t \in \mathbb{R}^{20\times 20}$, and $\boldsymbol{\theta}^* = (0.4, 0.7)$ are the generating parameters. We assume uniform priors $\varphi \sim \mathcal{U}(0,1)$, $\tau \sim \mathcal{U}(0,1)$. We time-augment by using the product[4] of a Weisfeiler-Lehman (WL) kernel (Shervashidze et al., 2011) on graphs $\mathbf{A}_t$ and a Gaussian RBF kernel on the time-channel for κ:

$$\kappa\left((\mathbf{A}_t, t), (\mathbf{B}_s, s)\right) = \mathrm{WL}(\mathbf{A}_t, \mathbf{B}_s) \cdot \exp^{-\frac{\|t-s\|^2}{\sigma}}, \tag{3.23}$$

in which the initial labels for all nodes in each graph in all sequences is taken to be identically 1. Furthermore, we perform two iterations of the message-passing and hashing procedure, and use a vertex histogram kernel as the base kernel.

We show the posterior we obtain from Algorithm 4 – using $N = 10^5$, $M = 250$, and the signature distance (3.1) using (3.23) as the static kernel κ – in Figure 3.7. From this we see that the S-ABC posterior has been able to concentrate significantly around the generating parameters $\boldsymbol{\theta}^*$, suggesting that our signature-based approach can furthermore be successfully applied to simulators generating data evolving in more general topological spaces than $\mathbb{R}^d$.

3.3 Discussion & conclusion

In this chapter, we introduced two novel approaches—Signature ABC and Signature Regression ABC—to performing approximate Bayesian computation with time series simulation models. Each method relies on the path signature—an object that is fundamental to the theory of controlled differential equations and rough paths—and that is associated with the path traversed by a sequence of data points. In particular, we make use of the recently developed signature kernel to construct and compute discrepancies between time series data arising in ABC settings without manually contriving summary statistics.

We show that the natural notion of distance between time series to which such an approach leads satisfies conditions under which the ABC posterior converges to the ground-truth posterior (under certain regularity conditions on the simulator's likelihood function) and discuss the robustness properties of the Signature ABC posterior as the number of data points $n \to \infty$ within a finite time horizon for a fixed ABC

[4]Such tensor product kernels are valid kernels on product spaces.

tolerance parameter. As an illustration of our proposed methods, we present multiple examples of Bayesian inference tasks in which our approaches outperform existing techniques that are common in the approximate Bayesian inference literature; indeed, in each experiment we consider, at least one signature-based method uniformly dominates competing methods across all three of the metrics considered in this chapter. We demonstrate that our methods flexibly accommodate a number of potentially helpful transformations of the data—for example, delay transformations—and in our final examples that our methods are applicable to more complex settings than univariate time series, for example multivariate and irregularly sampled sequences and even simulators that generate non-Euclidean time series.

While we have compared the different distance measures using a basic rejection algorithm in this chapter in order to allow for a simple and transparent comparison, we note that our proposed methods can be embedded within other more sophisticated sampling algorithms, for example MCMC or sequential Monte Carlo methods. Additionally for the Signature Regression ABC method, there is the possibility of incorporating mechanisms for generating more accurate regression results, for example using a pilot run to determine regions of non-negligible posterior mass as described in Fearnhead and Prangle (2012). There is also the possibility of pooling information from our signature-based ABC posteriors with summary-statistic-based ABC posteriors; this general idea is explored recently in Frazier et al. (2022), in which the authors consider posteriors of the form

$$\pi_{\text{ABC}}(\boldsymbol{\theta} \mid \mathbf{y}) = \omega \cdot \tilde{\pi}(\boldsymbol{\theta} \mid \mathbf{s}(\mathbf{y})) + (1 - \omega) \cdot \tilde{\pi}(\boldsymbol{\theta} \mid \mathcal{D}), \tag{3.24}$$

where $\omega \in [0, 1]$ is a weighting parameter, $\tilde{\pi}(\boldsymbol{\theta} \mid \mathbf{s}(\mathbf{y}))$ and $\tilde{\pi}(\boldsymbol{\theta} \mid \mathcal{D})$ are the ABC posteriors derived from a summary statistic-based and discrepancy-based approach to ABC, respectively, and $\pi_{\text{ABC}}(\boldsymbol{\theta} \mid \mathbf{y})$ is the resultant, "pooled" ABC posterior. The authors consider a decision-theoretic approach to the optimal choice of ω, which delivers guaranteed inferential accuracy under certain regularity conditions. Each of these extensions to our work may allow for improved approximations to the true posterior density.

3.3.1 Computational cost

While we have seen significant improvements upon baseline methods using our signature-based approaches, we also observe a drawback in the sense that our signature-based

approaches tend to incur a larger computational cost, at least with current implementations. This is reflected in Table 3.1, in which we tabulate approximate average CPU times required for each experiment described above for each method. Our signature-based methods are listed in the first three columns, and can be seen to incur higher CPU times in general.

However, we submit that the increased computation time associated with our signature-based approaches that we observe here are limited in their impact. The first reason for this is that ABC tends to incur a large computational cost in the first place, since many hundreds of thousands of simulations from the model are typically necessary. It is therefore not typically the case that computational resources will be very constrained in settings in which ABC is a feasible inference procedure. Secondly, it is currently difficult to assess the inherent computational burden associated with our signature-based approaches and to disentangle this from costs derived from implementations of the signature computations that are currently potentially suboptimal, due to the relatively embryonic nature of the use of signature methods in computational statistics and machine learning. It is indeed plausible that more efficient implementations of signature computations will emerge with time, given that research on signature methods in machine learning and computational statistics is active and relatively nascent. Thirdly, as discussed in Section 3.1.3, the computational complexity of signature kernel evaluations has a favourable scaling compared to popular alternative approaches to discrepancy-based ABC such as WASS, meaning that the difference in computational cost will change in favour of our signature-based approaches as data becomes larger than the data considered in the experiments presented in this Chapter. Finally, there exists some steps that can be taken but that we have not taken here to increasing the speed of computations, such as employing GPUs and using a truncated signature kernel as in Kiraly and Oberhauser (2019) – using the `KSig` package[5], for example[6]. Nonetheless, under certain circumstances – such as circumstances in which only a limited, fixed amount of wall-clock time is available to collect as many ABC posterior samples as possible – the additional computational cost under certain regimes associated with the signature-based methods we present may be detrimental, and it is possible that the resulting increase in Monte Carlo error may not balance out any increased accuracy seen through the use of signatures where they are the more appropriate tool.

[5] https://github.com/tgcsaba/KSig

[6] Of course, using a truncated signature kernel introduces a further complication of deciding on a truncation degree, the optimal value of which may be difficult to determine.

Experiment	Method						
	S-ABC	S-ABC (delay)	SR-ABC	WASS	WASS (delay)	SA-ABC	MMD
Ricker	2×10^2	2×10^2	2×10^4	6×10^1	8×10^1	10^2	4×10^1
GBM	10^3	10^4	6×10^4	4×10^3	4×10^3	9×10^2	4×10^3
B&H	10^4	10^4	5×10^4	2×10^2	2×10^2	2×10^1	6×10^1
GSE	10^4	–	–	2×10^2	–	–	10^5

Table 3.1: (CPU times) Approximate average CPU times (in seconds) for each ABC approach for constant simulation budgets and hardware availability. The Brock & Hommes model is listed as "B&H" to satisfy constraints on space.

3.3.2 Future work

Throughout the above, we have assumed that only one sequence $\mathbf{y}$ has been observed from the real world. This is a realistic assumption in many useful real-world cases; for example, this is often the case in macroeconomics or during a pandemic, where it would be incorrect to treat signals recorded at e.g. the country level as being *iid* rather than as different channels in a single observed sequence.

However, there are certain realistic settings in which multiple sequences $\{\mathbf{y}^{(j)}\}_{j=1}^{J}$ are recorded in which an *iid* assumption is reasonable. For example, in healthcare settings, recordings of patients with similar medical profiles may reasonably be modelled as *iid* draws from some underlying distribution. Similarly, in the natural or behavioural sciences, it is sometimes possible to perform multiple trials or repetitions of experiments in which the evolution of some quantity is recorded. In these cases, the following two generalisations of the approach taken in this chapter may be useful:

1. taking $\mathcal{P}^J = \{\mathbf{y}^{(j)}\}_{j=1}^{J}$, we may use the discrepancy measure

$$\begin{aligned}\mathcal{D}_{\text{S}}(\delta_{\mathbf{x}}, \mathcal{P}^J) &:= \left\|\mathbb{E}_{\mathbf{x}\sim\delta_{\mathbf{x}}}[\text{Sig}(\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\mathcal{P}}[\text{Sig}(\mathbf{y})]\right\|^2 \\ &= k(\mathbf{x},\mathbf{x}) + \frac{1}{J(J-1)}\sum_{i\neq j} k(\mathbf{y}^{(i)},\mathbf{y}^{(j)}) - \frac{2}{J}\sum_{j=1}^{J} k(\mathbf{x},\mathbf{y}^{(j)}) \qquad (3.25)\end{aligned}$$

each time we query a new parameter $\boldsymbol{\theta}$ – where $\delta_{\mathbf{x}}$ is a point mass located on $\mathbf{x} \sim p(\cdot \mid \boldsymbol{\theta})$ – as the distance measure in ABC. This provides a meaningful

comparison between a single output from the dynamic, stochastic simulator and the empirical measure on sequences given by the real-world dataset when the simulation budget should be kept as low as possible;

2. more generally, when there is greater tolerance for a larger simulation burden, one may instead simulate $N \geq 1$ times at each $\boldsymbol{\theta}$ to construct an empirical measure $\mathcal{P}_{\boldsymbol{\theta}}^N = \{\mathbf{x}^{(n)}\}_{n=1}^N, \mathbf{x}^{(n)} \overset{iid}{\sim} p(\cdot \mid \boldsymbol{\theta})$ and use the full MMD between (in general non-Dirac) measures on sequences:

$$\begin{aligned}\mathcal{D}_{\text{M}}(\mathcal{P}_{\boldsymbol{\theta}}^N, \mathcal{P}^J) &:= \left\| \mathbb{E}_{\mathbf{x}\sim\mathcal{P}_{\boldsymbol{\theta}}^N}[\text{Sig}(\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\mathcal{P}^J}[\text{Sig}(\mathbf{y})] \right\|^2 \\ &= \frac{1}{N(N-1)} \sum_{i \neq j} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \frac{1}{J(J-1)} \sum_{i \neq j} k(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) \\ &\quad - \frac{2}{NJ} \sum_{i,j} k(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}). \end{aligned} \tag{3.26}$$

The latter of these may also be useful in the case of a single observation and simulation with $J = N = 1$ in the following way: if the data-generating process is known to be ergodic, it may be reasonable to treat successive blocks/sub-sequences of $\mathbf{y}$ and $\mathbf{x}$ as being approximately *iid*. Then, taking $\mathcal{P}_{\boldsymbol{\theta}}^N$ and $\mathcal{P}^J$ to be the empirical measures associated with the collection of blocks of $\mathbf{x}$ and $\mathbf{y}$, respectively, $\mathcal{D}_{\text{M}}$ provides a reasonable discrepancy to be used in ABC for dynamic, stochastic simulation models with intractable likelihood functions.

Chapter 4

Deep Signature Statistics for Simulation-based Inference with Time-series Simulators[1]

As discussed in Section 1.5, dealing appropriately with time-series data of different kinds in SBI remains a challenge. In particular, the selection of an appropriate, low-dimensional set of summary statistics is both a non-trivial task and key to the quality of inference. We noted that some of the more recent approaches to SBI, such as neural DRE, are able to automate the learning of summary statistics by leveraging the expressiveness of neural networks. Yet, it is not guaranteed that the summary statistics implicitly generated through the use of such neural networks provide representations of sufficient quality for posterior inference. Fully connected networks, for example, still lack the inductive biases to easily extract meaningful representations from time-series. Indeed, as we will see in the experiments presented below, hand-crafted summary statistics can often outperform neural networks to a significant degree, raising the question of how automated techniques can fill the knowledge gap of domain expertise.

In this section, we motivate and present the novel use of the so-called "deep signature method" (Morrill et al., 2020; Kidger et al., 2019) for extracting features from multivariate sequential data for the purpose of performing LFI. The central object of study – the *path signature* – is, in a sense, a canonical feature transformation in that

[1]This chapter is based on Dyer et al. (2021b), which is joint work with Patrick Cannon and Sebastian M. Schmon. Horatio Boedihardjo generously provided the proof for Proposition 9 by email.

the signature of path-valued random variable captures all possible nonlinear effects. For this reason, one might expect to be able to generate better posterior estimates with a lower number of parameters or smaller sample budgets. Applications of the signature method have produced promising results in a number of learning tasks, including character recognition (Xie et al., 2018), gesture recognition (Li et al., 2017a), and early identification of Alzheimer's from clinical data (Moore et al., 2019).

We term this approach to learning summary statistics for time-series simulators deep signature statistics (DSS). The main idea is to embed a deep signature model (Kidger et al., 2019) – in which the signature appears as a pooling operation in a neural network – into an existing neural density ratio estimation pipeline. The deep signature model then automates the learning of model-dependent summary statistics in tandem with neural density ratio estimation of the likelihood-to-evidence ratio. We theoretically base this idea on the fact that – as we will show – the *truncated* path signature is a sufficient statistic when the truncation is taken at a sufficiently large depth. We demonstrate the DSS framework on posterior estimation tasks for a selection of models, and compare its performance against standard hand-crafted and learned summary statistics.

4.1 Method: Deep Signature Statistics

It is clear from Proposition 7 that the full path signature is a sufficient statistic for appropriately augmented sequences in a Banach space. Being an infinite dimensional object, however, the full signature cannot be used as an explicit summary statistic. In the interest of using the signature as an explicit summary statistic for sequences/paths in a finite-dimensional Euclidean space $\mathbb{R}^d$, we present the following result, providing a finite guarantee on the expressiveness of the depth-N truncated signature transform, Sig_N.

Proposition 9. *Let* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ *be a stream of data generated by a model* $p(\mathbf{x} \mid \boldsymbol{\theta})$*, with* $\mathbf{x}_i \in \mathbb{R}^d$ *and fixed, known observation times* $t_1 < t_2 < \cdots < t_n$*. Assume a linear interpolation* $\mathcal{I}$ *and that the path has been time- and basepoint-augmented. Let* $\hat{\mathrm{Sig}}_N(\mathbf{x})$ *be the depth-*N *signature transform of the linearly interpolated* $\mathbf{x}$*. Then, the path can be reconstructed exactly from the* $\hat{\mathrm{Sig}}_{n-1}(\mathbf{x})$*, and*

$$\pi(\boldsymbol{\theta} \mid \hat{\mathrm{Sig}}_{n-1}(\mathbf{x})) = \pi(\boldsymbol{\theta} \mid \mathbf{x}).$$

Proof. As there is a basepoint augmentation, it is sufficient to determine the slope of each interval $[t_i, t_{i+1}]$.

Let $a(i,j)$ be the slope of channel i on the interval $[t_j, t_{j+1}]$. Given the signature to degree $n-1$, we have $(\mu(i,k))_{1\le i\le d, 1\le k\le n-1}$ given by the following for $k \ge 1$,

$$\begin{aligned}\mu(i,k) &:= \int_{0<s_1<\cdots<s_k<1} \mathrm{d}s_1 \ldots \mathrm{d}s_{k-1}\mathrm{d}x^i_{s_k} \\ &= \int_0^1 \frac{(s_k)^{k-1}}{(k-1)!}\mathrm{d}x^i_{s_k} \\ &= \frac{1}{(k-1)!}\int_0^1 (s_k)^{k-1}\left[x^i_{s_k}\right]' \mathrm{d}s_k.\end{aligned}$$

Integrating on each interval separately and using that the slope is constant on each interval, we have

$$\mu(i,k) = \frac{1}{k!}\sum_{j=2}^{n} a(i,j-1)\left(t_j^k - t_{j-1}^k\right)$$

We now show that the $a(i,j-1)$ are uniquely determined by the $\mu(i,k)$ or, equivalently, the $k!\mu(i,k)$.

In matrix form, and with $\Delta t_j^n := t_j^n - t_{j-1}^n$, we can rewrite the above equations as

$$\begin{pmatrix} \Delta t_2^1 & \cdots & \Delta t_n^1 \\ \Delta t_2^2 & \cdots & \Delta t_n^2 \\ \vdots & & \vdots \\ \Delta t_2^{n-1} & \cdots & \Delta t_n^{n-1} \end{pmatrix}\begin{pmatrix} a(i,1) \\ a(i,2) \\ \vdots \\ a(i,n-1) \end{pmatrix} = \begin{pmatrix} 1!\mu(i,1) \\ 2!\mu(i,2) \\ \vdots \\ k!\mu(i,n-1) \end{pmatrix}$$

which for brevity we will write $(\Delta\mathbf{t})\mathbf{a} = \boldsymbol{\mu}$. A unique solution exists for the $a(i,j)$ if and only if

$$\det(\Delta\mathbf{t}) \neq 0. \tag{4.1}$$

Notice that since $t_1 = 0$, the first column of $\Delta\mathbf{t}$ simplifies to $(t_2, t_2^2, \ldots, t_2^{n-1})^\top$. Using this, and the fact that the determinant is unaffected by adding together columns, (4.1) is equivalent to

$$\det\begin{pmatrix} t_2 & \cdots & t_n \\ t_2^2 & \cdots & t_n^2 \\ \vdots & & \vdots \\ t_2^{n-1} & \cdots & t_n^{n-1} \end{pmatrix} \neq 0.$$

An equivalent requirement is

$$t_2 \cdots t_n \det \begin{pmatrix} 1 & \cdots & 1 \\ t_2 & \cdots & t_n \\ \vdots & & \vdots \\ t_2^{n-2} & \cdots & t_n^{n-2} \end{pmatrix} \neq 0.$$

The determinant here is of (the transpose of) a Vandermonde matrix. A standard result gives the determinant as (the reciprocal of) $\prod_{2 \leq i < j \leq n}(t_j - t_i)$ which is non-zero since $t_1 < t_2 < \cdots < t_n$. Therefore, $\boldsymbol{a}$ is uniquely determined by $\boldsymbol{\mu}$ as claimed.[2] □

In summary, only a finite number of terms of the signature are required to form a sufficient statistic of time-series data evolving in a finite-dimensional Euclidean space with known observations times. For this reason, we might expect that using the truncated signature representation of a time-series may therefore provide a useful and expressive basis from which we may learn parameter posteriors for time-series simulators. In general, however, the number of terms in the signature to degree $n-1$ is larger than n itself: note that the number of signature terms to degree m is given by $(d^{m+1} - 1)/(d - 1)$ for a d-dimensional path. While the dimensionality may be reduced further by making use of the more parsimonious log-signature, truncating to a low degree is necessary in order for the number of terms to be manageable. While the information loss this brings about may be tolerable in some instances, in others it may be detrimental to the inference task at hand.

In this chapter, we consider that this information loss may however be mitigated with the use of *deep signature transforms* (Kidger et al., 2019). A deep signature transform, shown in Figure 4.1 and given by

$$\sigma^{\varphi}(\mathbf{x}) = \left(f^{\varphi_{k+1}} \circ B_{N_k}^{\varphi_k} \circ \cdots \circ B_{N_2}^{\varphi_2} \circ B_{N_1}^{\varphi_1}\right)(\mathbf{x}) \tag{4.2}$$

entails repeated application of *blocks* $B_{N_i}^{\varphi_i}(\mathbf{x}) = \left(\mathrm{Sig}_{N_i} \circ \ell^i \circ \Phi^{\varphi_i}\right)(\mathbf{x})$ of three key elements: an augmentation of the stream with a learnable, stream-preserving feature map Φ^{φ_i}; a lift operation ℓ^i, which transforms the augmented stream into a stream of streams; and the depth-N_i truncated signature transform Sig_{N_i} applied to each substream, giving a stream of signatures. Blocks are by design able to concatenate, and after as many blocks as desired have been concatenated, the output is obtained by passing the output of the final block through an optional additional neural network

[2] We are indebted to Horatio Boedihardjo, who generously provided us with this proof by email.

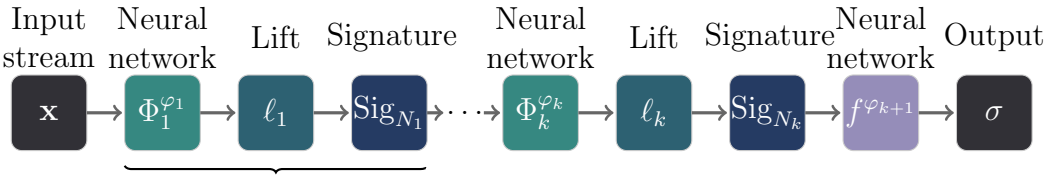


Figure 4.1: Deep signature transform with parameters $\varphi_1, \ldots, \varphi_{k+1}$.

$f^{\varphi_{k+1}}$. The ultimate effect is to capture higher order signature information using fewer terms (Chevyrev and Oberhauser, 2018; Kiraly and Oberhauser, 2019). In the above, $\varphi = (\varphi_1, \ldots, \varphi_{k+1})$. We provide further details on deep signature transforms in Appendix B.

Combining path signatures—with their strong mathematical basis—with the expressivity of neural networks has been seen to produce competitive results in a number of learning tasks (Kidger et al., 2019; Morrill et al., 2020). In this way, despite truncation, the use of a deep signature transform in place of the large signature to degree $n-1$ may yield approximately sufficient statistics. This makes it an interesting and potentially powerful candidate for use in likelihood-free inference settings as a means for generating model-specific summary statistics, which we term *deep signature statistics* (DSS).

4.2 Experiments

In the following, we compare the performance of DSS against state-of-the-art models for learning summary statistics from sequential data, and common hand-crafted summary statistics. In particular, for each summary statistic-learning method, we train the summary network as an embedding network, such that the summary network weights are optimised in tandem with a neural density ratio estimator and using the same loss function. The composite embedding network-classifier network then approximates the likelihood-to-evidence ratio, or equivalently the posterior-to-prior ratio $\pi(\boldsymbol{\theta} \mid \sigma^{\varphi}(\mathbf{x}))/\pi(\boldsymbol{\theta})$. We compare against PENs and a recurrent neural network (RNN) trained as an embedding network[3]. We do this for two models: a discretized

[3] Wiqvist et al. (2019) did not consider training PEN as an embedding network in their original work, but we adopt this approach here in order to ensure a fair comparison.

Ornstein-Uhlenbeck process, which is Markovian and has a tractable likelihood; and the Ricker model, a further Markovian model with an intractable likelihood.

4.2.1 Neural network specifications

Deep signature statistics The deep signature model we use involved three neural-lift-signature blocks followed by a final recurrent network. The neural component of the first block consisted of a feedforward network with kernel size 3 and 2 hidden layers of size 16 swept across the input stream. The output size of this network was 3, so that initial layer augmented the input stream with an additional 3 channels. The neural components of the remaining two blocks were recurrent networks with 2 hidden layers of size 16. For each block, we use expanding windows with initial size 2 that grew by 1 time step in each iteration, followed by the signature transform truncated at degree 3. For all simulators, we apply basepoint and time augmentations to the input stream before passing it through the deep signature model, and take an output of size 3. This yields a model with 9,735 trainable parameters.

Partially exchangeable networks For our experiments, we follow Wiqvist et al. (2019) and take the ϕ network to be a fully connected network with three layers of sizes 11, 100, and 50 and output size 10, and the ρ network to be a fully connected network with four layers of sizes $(10+r)$, 50, 50, and 20. ReLU activations were used for all hidden layers. For PEN1, this yields a model with 10,093 trainable parameters.

Recurrent neural network The recurrent network model consists of two recurrent neural networks. The first network has layers of size 64, 64, and 32, with an output of size 6, while the second layer has layers of size 32, 32, and 32 with output size 7. Windows of size 4 were swept across the input for both networks, with strides of 4 and 2 in the first and second, respectively. Altogether, this yields a model with 10,157 trainable parameters.

4.2.2 Evaluation metrics

To assess the quality of the estimated posteriors, we compute the sliced Wasserstein distance (SWD) (Peyre and Cuturi, 2019) between samples from the approximate

ground-truth posterior and samples from the estimated posterior densities. In all cases, SWDs were computed using the Python Optimal Transport package (Flamary et al., 2021) and 1000 posterior samples from the posterior density estimated in each training round. To train the ratio estimator, we consider the sequential training approach described in Section 1.5.4.4 in which we generate 1000 training examples during each round for 20 rounds.

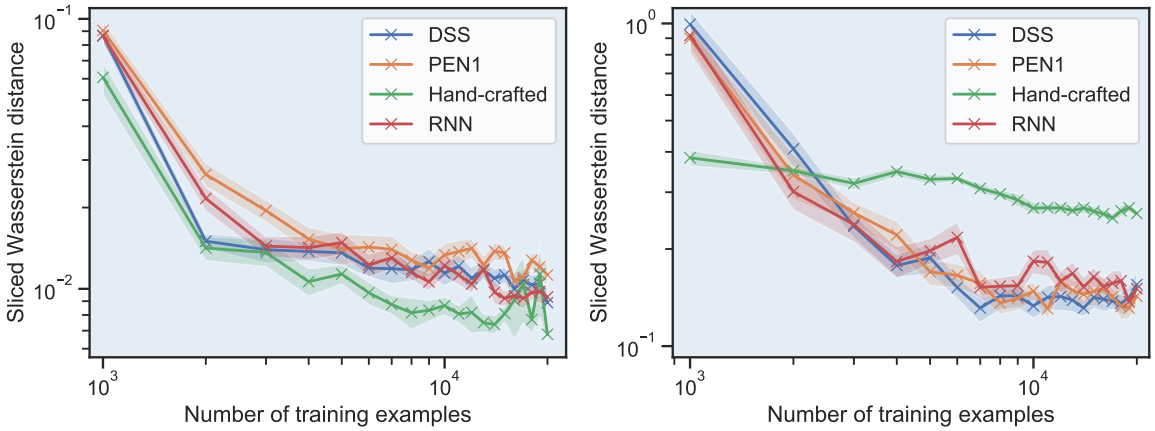
4.2.3 Ornstein-Uhlenbeck process

We consider here the OU model and inference task as described in Section 1.7.5.

A quantitative evaluation of the quality of the estimated posteriors is shown in Figure 4.2a. Here, we compare the SWD between samples from the true posterior and estimated posteriors for DSS, hand-crafted summary statistics, RNN, and PEN at each training round. The hand-crafted summaries we used were the mean, standard deviation, and autocorrelations at lags 1 and 2 of the observed time-series, giving four hand-crafted summary statistics.

Of the learned summaries, DSS tends to perform at least as well as RNN and PEN in the majority of training rounds, while it significantly outperforms both for an intermediate number of training examples. In particular, DSS matches or exceeds the performance of RNN and PEN for 13 out of 20 rounds. Furthermore, while all methods perform relatively poorly during the first round, DSS improves rapidly thereafter, with RNN and PEN requiring 4-5 times as many training examples to match the performance of DSS in its second round.

The hand-crafted summary statistics deserve further discussion. For this simulator, we observe that the hand-crafted summary statistics outperform all learned summaries at almost every training round. This demonstrates the importance of well-chosen summary statistics and inductive biases, and the non-trivial nature of learning appropriate summary statistics for time-series data: even with state-of-the-art neural network models such as RNN and PEN for summarizing time-series data, it is difficult to meet, let alone surpass, the performance of sensible hand-crafted summaries. Using the inductive biases introduced by the deep signature model, we have been able to close this gap at an early stage in the training procedure, but it is also evident that more research is required to fully match performance across all experiment settings.



(a) Ornstein–Uhlenbeck **(b)** Ricker model

Figure 4.2: The sliced Wasserstein distances between the true and estimated posterior densities for each summary statistic method at each training round for the (a) Ornstein–Uhlenbeck process and (b) Ricker model. Crosses and shaded regions indicate mean and standard error over 20 different seeds.

4.2.4 Ricker model

We consider here the Ricker model and inference task as described in Section 1.7.1.

In Figure 4.2b, we show the SWD between the samples from the approximate ground-truth posterior and estimated posteriors for each summary statistic method. Samples from the approximate ground-truth posterior density $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ for the Ricker model were obtained using particle MCMC (Andrieu et al., 2010) following the guidelines of Schmon et al. (2021). The hand-crafted summary statistics used in this instance are those proposed in Wood (2010), and consist of: the autocovariances to lag 5; the mean; the number of zeros in the sequence; the coefficients of the regression $\mathbf{x}_{t+1}^{0.3} = \beta_1 \mathbf{x}_t^{0.3} + \beta_2 \mathbf{x}_t^{0.6} + \epsilon_t$ for error term ϵ_t; and the coefficients of the cubic regression of the ordered differences $\mathbf{x}_t - \mathbf{x}_{t-1}$ on their observed values.

From Figure 4.2b, we see that DSS matches or exceeds PEN's (resp. RNN's) performance in 14 (resp. 10) out of 20 rounds. We also see that, while each learned summary performs comparatively poorly until the third round, DSS leads for intermediate numbers of training examples: both PEN and RNN require between 1000-4000 additional training examples to match DSS's performance in a number of instances. This example also highlights the possibility that learned summary statistics can outperform expert hand-crafted summaries, in particular, when model complexity doesn't allow for straightforward selection.

4.3 Discussion

In the previous subsection, we presented experiments on the use of a deep signature model as an embedding network in neural density ratio estimation for time-series models. The aim of this investigation was to assess whether this approach, termed DSS, provides a competitive means in practice to automating the task of learning summary statistics and density ratios concurrently for likelihood-free Bayesian inference. We motivated this approach as being one that may provide approximately sufficient statistics for approximate Bayesian inference, and we compared this approach to alternative approaches for time-series models, including using a PEN and a high-capacity RNN as alternative embedding networks. While we observed some minor improvements with the use of DSS, the performance was not significantly improved over the considered baselines. In addition, each method incurred a comparable computational expense: on the same hardware, all 20 rounds (with 1000 training examples in each round) for both the OU and Ricker models required between 2-3 hours for DSS and RNN, while PEN was slightly slower at 4-5 hours. There is therefore also little to distinguish these methods in terms of observed computational expense.

In the following chapter, we consider an alternative regime in which signatures may provide a more significant advantage. In the experiments we present with DSS, we consider a moderately high simulation budget of 2×10^4 (20 rounds of 1000 simulations each). For such simulation budgets, learning good summary statistics may be achievable for many simulators with a wide variety of embedding networks. However, the appeal of the path signature is that the features they provide are ready-made and expressive. It may therefore follow that the signature enables more accurate density ratio estimation at lower simulation budgets than neural density ratio estimation methods in which summary statistics must be learned. We consider this possibility in more detail in the following chapter.

4.4 Acknowledgements

We gratefully acknowledge Horatio Boedihardjo, who provided us with the proof for Proposition 9 by email in January 2021.

Chapter 5

Density Ratio Estimation with Signature Kernel Logistic Regression[1]

Neural methods for estimating the likelihood function (Papamakarios et al., 2019), posterior density (Greenberg et al., 2019), or likelihood-to-evidence ratio (Hermans et al., 2020) as discussed in the previous section, have been seen to perform competitive likelihood-free inference with far fewer samples than are typically required in more traditional approaches such as approximate Bayesian computation (ABC) (Lueckmann et al., 2021). However, despite the greater sample efficiency provided by these approaches, their budget requirements can still be too high for very complex models, for example high-dimensional spatio-temporal simulations. Indeed, a single call to a simulator can take hours to days for multi-scale models of 3D tumour growth (e.g. Jagiella et al., 2017) or multiple thousands of CPU hours for climate models (e.g. Danabasoglu et al., 2020). For others, high simulation budgets may in principle be attainable but undesirable due to the concomitant financial and environmental costs.

With this challenge in mind, we propose and test a kernel method for density ratio estimation for time-series simulators in this section. It is well known that kernel methods are useful learning tools in low-training-example regimes, providing rich, ready-made data representations (Shawe-Taylor and Cristianini, 2004). Moreover, the signature can extract powerful features from time-series data, acting analogously to moment-generating functions for path-valued random variables. To benefit from the

[1]This chapter is based on Dyer et al. (2022c), which is joint work with Patrick Cannon and Sebastian M. Schmon.

advantages of both kernels and the signature, we present an approach to LFI based on this signature kernel, demonstrating more accurate inferences than competing density ratio techniques when the simulation budget is limited.

5.1 Method: SignatuRE

Our goal is to perform amortised density ratio estimation as described by Hermans et al. (2020) and in Section 1.5.4.1. In particular, we seek to be able to do so in low-simulation-budget enivronments. As we will see in experiments below, learning both summary statistics and a classifier can be challenging in such regimes. To ameliorate this, we propose to build a classifier that leverages the signature kernel, which defines a universal kernel for multivariate and possibly irregularly sampled sequential data. The core idea is that using the predefined features captured by the signature and made available by the signature kernel may yield a more reliable density ratio estimator in low-sample regimes than alternative methods for which summary statistics must be learned.

To construct a probabilistic binary classifier using the signature, we may use the fact that a third kernel m on $\mathcal{X}^n \times \mathbf{\Theta}$ can be composed given two kernels $k : \mathcal{X}^n \times \mathcal{X}^n \to \mathbb{R}$ and $l : \mathbf{\Theta} \times \mathbf{\Theta} \to \mathbb{R}$ as

$$m\left((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}')\right) = k\left(\mathbf{x}, \mathbf{x}'\right) l\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right). \tag{5.1}$$

Taking k to be the signature kernel (2.12) and l to be a standard universal kernel on $\mathbf{\Theta}$, we may construct a kernel-based binary classifier for the purpose of performing DRE for expensive time-series simulators, in this sense bypassing the need to learn summary statistics in addition to a density (ratio) estimator.

For a regularisation constant $\lambda \in \mathbb{R}_+$, training a kernel binary classifier with loss ℓ amounts to solving the optimisation problem

$$\min_{f \in \mathcal{H}_m} \sum_{i=1}^{n} \ell\left(f(\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)}), z_i\right) + \frac{\lambda}{2} \|f\|_{\mathcal{H}_m}^2, \tag{5.2}$$

where $\mathcal{H}_m$ is the RKHS associated with m. By the Representer Theorem, the solution to (5.2) is of the form

$$f\left(\mathbf{x}, \boldsymbol{\theta}\right) = \sum_{i=1}^{n} a_i k\left(\mathbf{x}^{(i)}, \mathbf{x}\right) l\left(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}\right) \tag{5.3}$$

for real coefficients a_i. Throughout, we use the logistic loss as ℓ, since this is known to yield classifiers with well-calibrated probability estimates. This approach to learning the likelihood-to-evidence ratio is appealing since m is a universal kernel:

Proposition 10. *Let $\mathcal{X}^n$ be a compact set of length-n sequences in $\mathbb{R}^p$ with $1 \leq n, p < \infty$. Assume that $\forall \mathbf{x} \in \mathcal{X}^n$, $\mathbf{x}$ has at least one monotone coordinate and $\mathbf{x}_0 =$ constant. Also let $k : \mathcal{X}^n \times \mathcal{X}^n \to \mathbb{R}$ be the discretised signature kernel (which operates on sequences $\mathbf{x} \in \mathcal{X}^n$) and l be a universal kernel on a compact subset $\boldsymbol{\Theta}$ of $\mathbb{R}^d$. Then m as defined in Equation (5.1) is a universal kernel on $\mathcal{X}^n \times \boldsymbol{\Theta}$.*

Proof. From Kiraly and Oberhauser (2019, Theorem 1), the (discretised) signature kernel is a universal kernel on the space of fixed-length sequences $\mathcal{X}^n$. Since $\mathcal{X}^n$ and $\boldsymbol{\Theta}$ are compact and l is universal on $\boldsymbol{\Theta}$, Blanchard et al. (2011, Lemma 5.2) applies the desired result follows. □

The universality of m then enables learning an arbitrarily accurate ratio estimate.

5.1.1 Low-rank approximation

Computing the signature kernel for all pairs $\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)$ in the simulated dataset can be expensive if the $\mathbf{x}^{(i)}$ are long and/or are high-dimensional. Therefore, we use the kernel m defined in (5.1) – with k the signature kernel and $l : \boldsymbol{\Theta} \times \boldsymbol{\Theta} \to \mathbb{R}$ an anisotropic Gaussian RBF kernel – and the Nyström approximation to first find a representation of each pair $(\mathbf{x}, \boldsymbol{\theta})$ before feeding this low-dimensional approximation into the logistic regression model. For a given kernel m, the Nyström method (Williams and Seeger, 2001; Yang et al., 2012) provides a low-dimensional approximation $\hat{\phi}$ of the high- or potentially infinite-dimensional feature map $\phi(a) := m(a, \cdot)$ as follows: assume the kernel m is of rank q such that for any data $\{a^{(i)}\}_{i=1}^N$ we may write the corresponding Gram matrix $\mathbf{K}$ as

$$\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}^T, \tag{5.4}$$

where $\mathbf{U} \in \mathbb{R}^{N \times q}$ is the matrix of eigenvectors and $\mathbf{D} = \operatorname{diag}(\lambda_1, \ldots, \lambda_q) \in \mathbb{R}^{q \times q}$ is the diagonal matrix consisting of eigenvalues λ_i. Then denoting the first q rows of $\mathbf{U}$ as $\mathbf{U}_q$, we may find an approximate feature representation of $\mathbf{y}$ under m as Yang et al. (2012)

$$\hat{\phi}(a) = \mathbf{D}^{-\frac{1}{2}}\mathbf{U}_q^T \left[m\left(a, a^{(1)}\right), \ldots, m\left(a, a^{(q)}\right)\right]^T. \tag{5.5}$$

Using these approximate feature representations obtained with the Nyström approximation, we then construct a linear logistic regression model by solving the following optimisation problem:

$$\min_{\mathbf{w}\in\mathbb{R}^q} \sum_{i=1}^{n} \ell\left(\mathbf{w}^T \hat{\phi}\left(a^{(i)}\right), z_i\right) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{5.6}$$

where ℓ is the logistic loss. We omit the use of an intercept in the linear logistic regression optimisation problem above for simplicity, but include it in practice.

Throughout the rest of this chapter, we term this approach to performing ratio estimation with the signature kernel and logistic regression SignatuRE.

5.2 Experiments

In this section, we present experiments on the relative performance of the SignatuRE method against possible alternatives for DRE in likelihood-free inference contexts. For each task, we compare the quality of the posterior estimated with SignatuRE against the posteriors estimated with three alternatives:

1. a neural network consisting of a combination of a GRU model and ResNet classifier (GRU-ResNet). The gated recurrent unit (GRU) has trainable parameters ϕ and consists of two stacked GRU layers of size 32. The GRU and residual neural network (ResNet) are trained concurrently with the same cross-entropy loss, such that the GRU learns a low-dimensional summary $\mathbf{s}_\phi(\mathbf{x})$ as the ResNet learns the density ratio;

2. a ResNet which instead consumes predefined, hand-crafted summary statistics $\tilde{\mathbf{s}}(\mathbf{x})$ that are tailored to the inference task and known to be informative of the parameters to be inferred for tractable simulation models, or that are commonly used elsewhere in the literature when the simulation model is not tractable. Such an approach should be considered a gold standard that is not generally available for complex, opaque simulation models whose structure cannot be exploited to derive suitable summary statistics. We refer to this method as the Bespoke ResNet;

3. ratio estimation with double kernel logistic regression (K2-RE), a modification of K2-ABC (Park et al., 2016) that we propose as an alternative kernel-based method for DRE. The setup is identical to SIGNATURE with the exception that, instead of the signature kernel, we use

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\frac{\widehat{\mathrm{MMD}}^2\left(\mu_{\mathbf{x}}, \mu_{\mathbf{x}'}\right)}{\epsilon}\right) \tag{5.7}$$

as the positive definite kernel on $\mathbf{x}$, where $\mu_{\mathbf{x}}$ is the empirical measure consisting of the $n_{\mathbf{x}}$ points comprising $\mathbf{x}$ and

$$\widehat{\mathrm{MMD}}^2\left(\mu_{\mathbf{x}}, \mu_{\mathbf{x}'}\right) = -\frac{2}{n_{\mathbf{x}} n_{\mathbf{x}'}} \sum_{i=1}^{n_{\mathbf{x}}} \sum_{j=1}^{n_{\mathbf{x}'}} \chi\left(\mathbf{x}_i, \mathbf{x}'_j\right) + \frac{1}{n_{\mathbf{x}}\left(n_{\mathbf{x}} - 1\right)} \sum_{i=1}^{n_{\mathbf{x}}} \sum_{j \neq i} \chi\left(\mathbf{x}_i, \mathbf{x}_j\right) + \frac{1}{n_{\mathbf{x}'}\left(n_{\mathbf{x}'} - 1\right)} \sum_{i=1}^{n_{\mathbf{x}'}} \sum_{j \neq i} \chi\left(\mathbf{x}'_i, \mathbf{x}'_j\right) \tag{5.8}$$

is an unbiased estimate of the kernel maximum mean discrepancy between $\mu_{\mathbf{x}}$ and $\mu_{\mathbf{x}'}$ for an appropriate kernel χ (see Section 3. Park et al., 2016). We use a Gaussian RBF and the median heuristic (Section 4. Park et al., 2016) for χ.

For the static kernel κ in the signature kernel (see Chapter 2), we use a Gaussian RBF kernel with scale parameter chosen as $\mathrm{median}\{\|\mathbf{y}_i^* - \mathbf{y}_j^*\|_{i,j}^2\}$, where $\mathbf{y}^* = (\mathbf{y}_1^*, \ldots, \mathbf{y}_n^*)$ is the observation. For the kernel $l : \boldsymbol{\Theta} \times \boldsymbol{\Theta} \to \mathbb{R}$, we use an anisotropic Gaussian RBF kernel. To tune the length scale hyperparameters for l, the regularisation parameter λ, and the ϵ parameter for K2-RE, we use Bayesian optimisation and 5-fold cross-validation (see the Supplementary Material for further details.) To train the logistic regression models, we use the L-BFGS algorithm (Zhu et al., 1997) with a maximum number of 500 iterations.

To construct the set of negative examples $(\mathbf{x}, \boldsymbol{\theta}) \sim p\left(\mathbf{x}\right)\pi\left(\boldsymbol{\theta}\right)$ for SIGNATURE and K2-RE, we choose a proportion $K > 0$ of the $\mathbf{x}^{(i)}$ and pair them with some $\boldsymbol{\theta}^{(j)}$, $j \neq i$. $K > 1$ may also be chosen, in which case some $\mathbf{x}^{(i)}$ will appear multiple times in the set of negative examples. Unless stated otherwise, we take $K = 1$ and $q = B_{\min}(K + 1)$ in the Nyström approximation for both SIGNATURE and K2-RE, where $B_{\min}$ is the smallest simulation budget considered in the experiment. This value for q is chosen since it is the largest value that can be consistently applied across the range of simulation budgets considered in a given experiment.

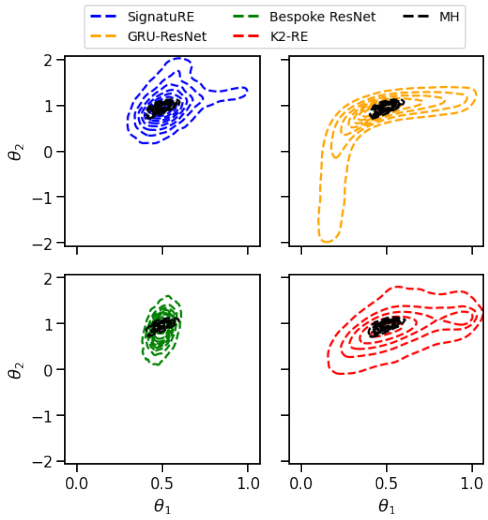


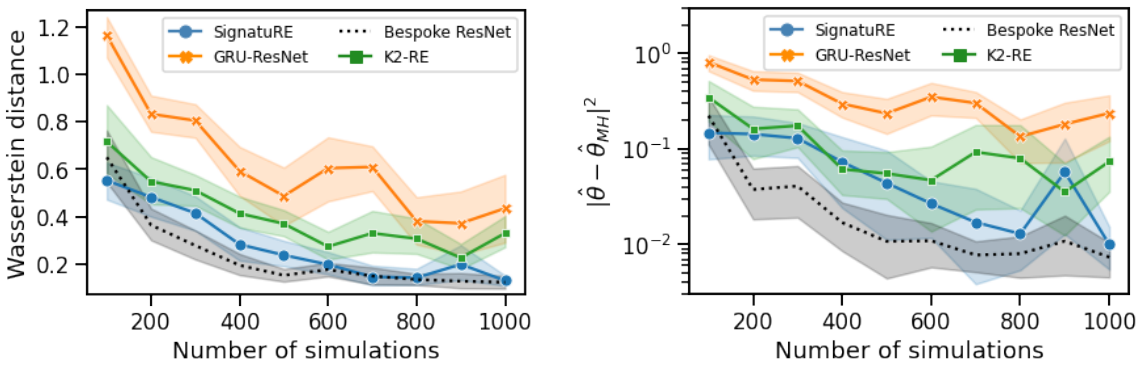
Figure 5.1: Ornstein-Uhlenbeck: Posteriors obtained with SignatuRE (blue, top left), GRU-ResNet (orange, top right), the Bespoke ResNet (Bespoke ResNet) (green, bottom left), and K2-RE (red, bottom right) for a budget of 500 simulations and the approximate ground truth posterior obtained using the true likelihood function and Metropolis-Hastings (black).

Remark 5. *While the methodology we have developed and will test in the following pages has been designed to address the computational expense associated with the cost of simulating sufficiently many times for existing* SBI *methods to work well, the experiments presented below do not make use of models that are expensive to run. The reason for this is that models that entail a genuinely large computational cost to simulate often lack tractable likelihood functions, preventing us from acquiring (a Monte Carlo approximation to) a ground-truth posterior density (for example through* MCMC*). Without this (approximate) ground-truth, it is difficult to assess the performance of the methods. Instead, we focus in this Section on designing scenarios that were* "as if" *the models were expensive to simulate by severely limiting the simulation budget that the* SBI *methods tested below are afforded.*

5.2.1 Ornstein-Uhlenbeck process

Here, we consider the OU model described in Section 1.7.5.

We compare SignatuRE against the alternative DRE methods described in Section 5.2. As $\tilde{\mathbf{s}}(\mathbf{x})$, we use the intercept and slope of a linear regression of $\mathbf{x}_t$ vs. $\mathbf{x}_{t-1}$ as estimated with least squares (i.e. the maximum likelihood estimate) and the mean value of $\mathbf{x}$. These estimate $\theta_1 \exp(\theta_2)\Delta t$, $1 - \theta_1 \Delta t$, and $\exp(\theta_2)$, respectively, and



(a) Wasserstein distances

(b) Euclidean distances between means

Figure 5.2: Ornstein-Uhlenbeck (a) Wasserstein distances between posteriors and (b) and Euclidean distances between posterior means (mean + 95% confidence intervals) obtained with each density ratio estimation method and the approximate ground truth posterior.

are thus informative summary statistics for $\boldsymbol{\theta}$. For GRU-ResNet, we apply a linear layer of size 3 after the GRU in order to match the dimension of $\tilde{\mathbf{s}}$, resulting in a GRU with 9,795 trainable parameters.

In Figure 5.1 we show contour plots obtained by pooling the samples obtained from each ratio estimation method with a simulation budget of 500 simulations across 20 different seeds. Samples from the approximate ground truth posterior, obtained with MH and the true likelihood function, are shown with black contour lines throughout. Additionally, we show in Figure 5.2a the Wasserstein distances (WD) between the estimated posteriors and the approximate ground truth posterior, and in Figure 5.2b the distances between the means of the estimated and approximate ground truth posteriors, for each ratio estimation method.

From this, we observe that GRU-ResNet (orange, top right of Figure 5.1) failed to learn both informative summary statistics and an accurate ratio estimator with a low simulation budget, despite the simplicity of the model. In contrast, an identical residual network used for Bespoke ResNet (green, bottom left of Figure 5.1) was able to learn a good estimate of the density ratio, even from such a limited simulation budget and with a summary statistic vector of identical size, but with the key difference that the summary statistics were predefined and designed to be informative of the parameter values being inferred.

This may be seen as an ablation study and suggests that the additional problem of learning summary statistics is the primary contributing factor to the relatively poor

performance of GRU-ResNet.

We also observe that, of the methods that do not use hand-crafted summary statistics, SignatuRE tends to exhibit superior performance. This is apparent from the posterior plots in Figure 5.1, and from Figure 5.2a in which SignatuRE consistently generates smaller WDs than GRU-ResNet and K2-RE and lags only slightly behind Bespoke ResNet.

From Figure 5.2b we see that SignatuRE tends to generate a significantly better parameter point estimate than GRU-ResNet and is additionally a slight improvement on K2-RE in this respect. The latter indicates that the success of SignatuRE in the low-simulation-budget regime is not only attributable to the expressive, pre-existing feature representations available with *general* kernel methods, but also to the fact that the *sequentialisation* of the kernel employed in SignatuRE captures important information on the time-dependence of the data whereas in K2-RE the data is treated as *iid*.

5.2.2 Moving average model

We next consider a simple moving average model of order 2 (MA(2)), for which the data-generating process given parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$ is

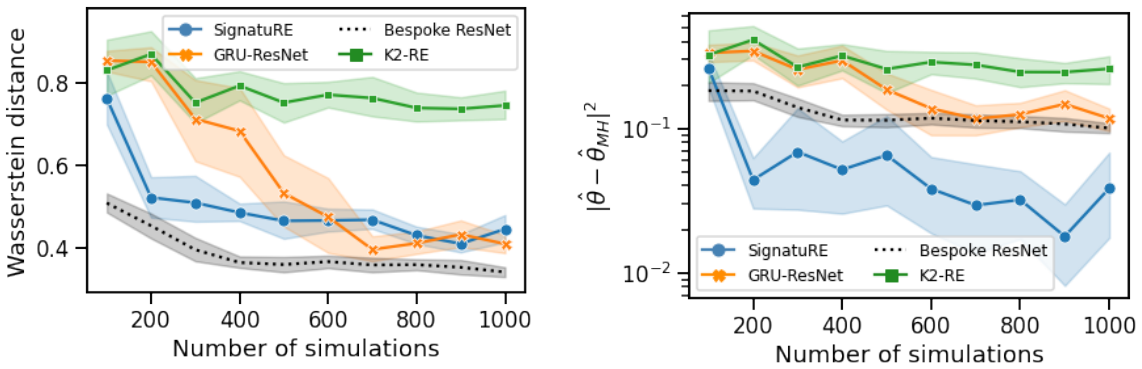
$$\mathbf{x}_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \in \mathbb{R}, \quad \epsilon_t \sim \mathcal{N}(0, 1). \tag{5.9}$$

We generate $\mathbf{x}^* \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*)$ with $\boldsymbol{\theta}^* = (0.6, 0.2)$ and consider the task of estimating $\pi(\boldsymbol{\theta} \mid \mathbf{x}^*)$ given a uniform prior over the triangle given by $\theta_1 + \theta_2 > -1$, $\theta_1 - \theta_2 < 1$, and $\theta_2 < 1$. Such a prior ensures that the model parameters are identifiable (Marin et al., 2012).

As $\tilde{\mathbf{s}}(\mathbf{x})$ we use the variance of the observed stream and the autocorrelations for lags 1 and 2. These give estimates of

$$\mathrm{Var}(\mathbf{X}) = 1 + \theta_1^2 + \theta_2^2, \quad \rho_1 = \frac{\theta_1 + \theta_1\theta_2}{1 + \theta_1^2 + \theta_2^2},$$
$$\text{and } \rho_2 = \frac{\theta_2}{1 + \theta_1^2 + \theta_2^2},$$

respectively, and are thus informative about $\boldsymbol{\theta}$. We once again apply a single linear layer of size 3 following the GRU in GRU-ResNet to match the dimensions of the summary statistics in Bespoke ResNet.



(a) Wasserstein distances

(b) Euclidean distances between means

Figure 5.3: MA(2) (a) Wasserstein distances between posteriors and (b) and Euclidean distances between posterior means (mean + 95% confidence intervals) obtained with each density ratio estimation method and the approximate ground truth posterior.

We show in Figure 5.3a the WDs between samples from the posteriors estimated with each density ratio estimation method and the approximate ground truth posterior obtained with Metropolis-Hastings MCMC. In Figure 5.3b, we show the Euclidean distances between the means of the posteriors estimated with the different density ratio estimators and the approximate ground truth posterior. In this experiment, we once more see that BESPOKE RESNET significantly outperforms GRU-RESNET in estimating the shape of the posterior distribution, despite the fact that they use identical residual networks to perform the density ratio estimation and that $\dim(\tilde{\mathbf{s}}) = \dim(\mathbf{s}_\phi)$. This again suggests that the complex task of learning summary statistics in addition to learning the density ratio is the source of the difference in their performance.

We further observe that SIGNATURE outperforms GRU-RESNET both in terms of the WD and distances between the estimated and approximate ground truth posterior means for simulation budgets of less than 500. For simulation budgets of 600-1000, SIGNATURE and GRU-RESNET display comparable performance according to the WDs, while SIGNATURE continues to obtain superior posterior mean estimates. Interestingly, SIGNATURE additionally yields better estimates of the posterior mean than BESPOKE RESNET, despite the fact that this density estimator has a considerable advantage through the use of hand-crafted summary statistics that are known to be informative of the parameters being inferred. As in the previous experiment, the success of SIGNATURE appears to be attributable not only to the general properties of kernel methods that make them appealing in low-sample regimes – their ready-made, expressive feature spaces – but also to the fact that the signature accounts for the

ordering of observations. We believe this explains the gap in performance between K2-RE and SignatuRE despite the former also being a kernel method.

5.2.3 Complex, intractable example: partially-observed stochastic epidemic

Finally, we consider a more complex example with an intractable posterior distribution. The model we consider here is a partially observed version of the generalised stochastic epidemic model described in Section 1.7.4: we simulate the model using the Gillespie algorithm (Gillespie, 1977) and observe the series $\mathbf{Z} = (X_{i\Delta t}, Y_{i\Delta t})_{i=0}^{D} \in \mathbb{R}^{2\times(D+1)}$ at regular time intervals of length $\Delta t = 0.5$ with $D = 100$. This defines the model's likelihood function $p(\mathbf{z} \mid \boldsymbol{\theta})$. We then consider the task of estimating $\pi(\boldsymbol{\theta} \mid \mathbf{Z} = \mathbf{z}^*)$ for $\mathbf{z}^* \sim p(\mathbf{z} \mid \boldsymbol{\theta}^*)$, $\boldsymbol{\theta}^* = (10^{-2}, 10^{-1})$, and priors $\beta \sim \Gamma(0.1, 2)$ and $\gamma \sim \Gamma(0.2, 0.5)$. To sample from the posterior in this case, we use a likelihood-free sampling-importance-resampling (SIR) scheme[2]: we sample $\mathcal{T} := \{\boldsymbol{\theta}_m\}_{m=1}^{M}$ from the prior, before resampling $\{\boldsymbol{\theta}'_m\}_{m=1}^{M'}$ from $\mathcal{T}$, where each sample in $\mathcal{T}$ has weight proportional to the density ratio estimated by the classifiers. We take $M = 5 \times 10^4$ and $M' = 10^3$.

In this instance, the ground truth posterior distribution is not available for comparison. For this reason, we assess the quality of inferences by comparing against the posterior obtained from sequential Monte Carlo ABC (SMC-ABC) (Beaumont et al., 2009) in which we use the Euclidean distance between time-series

$$\sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \tag{5.10}$$

as the distance measure with 10^7 simulations, distance $\|\mathbf{x} - \mathbf{x}^*\|^2$, Gaussian kernel, and ϵ decay factor equal to 0.8. We again compare SignatuRE with GRU-ResNet, Bespoke ResNet, and K2-RE. For Bespoke ResNet, we use the mean of each series, log variance of each series, autocorrelation coefficients for lags 1 and 2 of each series, and the cross-correlation coefficient between the two series as $\tilde{\mathbf{s}}(\mathbf{z})$, which are common summary statistics for stochastic kinetic models (Papamakarios et al., 2019; Greenberg et al., 2019).

[2]We found that the Metropolis–Hastings scheme adopted in the rest of this chapter performed poorly here.

We present the median Wasserstein distance between the estimated posteriors and the approximate ground truth posterior from SMC-ABC in Table 5.1, in which suffix "-5" indicates that $K = 5$ for kernel methods (otherwise $K = 1$ is used as before). Median values are obtained by repeating the inference procedure over 10 different random seeds. Our methods are either best (**bold**) or second-best (*italics*), but outperformed by GRU-ResNet at a budget of 1000 simulations.

5.3 Discussion

This section discusses the use of signature transforms via the signature kernel as automatic and effective feature extractors for likelihood-to-evidence ratio estimation. Our method, based on universal kernels and termed SignatuRE, delivers competitive performance even when sample numbers are very low. Indeed, our simulation studies suggest that using signatures as features improves upon a time-series specialised GRU-ResNet or kernels based on MMD in low-simulation-budget scenarios. In our experiments, SignatuRE was only consistently outperformed by a classifier that used bespoke hand-crafted summary statistics which were constructed by carefully inspecting the model structure. For real, complex simulators, such an approach is infeasible, making the proposed method appealing.

5.3.1 Computational expense

Empirically, we observe SignatuRE to entail a comparable computational cost to GRU-ResNet, the former typically requiring 3-5 CPU hours for training and inference and the latter typically requiring 1-2 CPU hours. For the simulation models for which we suppose our approach may be most helpful – those with significantly limited simulation budgets – we expect this to amount to a negligible difference: 1-4 additional CPU hours would allow for few or no additional complete simulations to be generated.

5.4 Future work

To extend the work presented in this chapter, it would be instructive to perform a more thorough theoretical investigation into DRE via SIGNATURE, or via kernel logistic regression more generally. In particular, it may be possible to derive explicit approximation errors on the target posterior density through the consideration of generalisation bounds for kernel logistic regression models. This will be the subject of future work.

Method	**Simulation budget**				
	50	**100**	**200**	**500**	**1000**
GRU-RESNET	0.434	0.425	0.355	0.273	*0.090*
K2-RE	*0.417*	0.432	0.407	0.454	0.431
K2-RE-5	0.440	0.427	0.374	*0.206*	0.255
SIGNATURE	0.430	*0.411*	*0.351*	0.513	0.321
SIGNATURE-5	**0.241**	**0.333**	**0.176**	**0.133**	**0.083**
BESPOKE RESNET	0.379	0.222	0.146	0.104	0.092

Table 5.1: Median Wasserstein distance from the SMC-ABC posterior for the partially-observed epidemic model (from 10 seeds). Of the methods that do not use hand-crafted, informative summary statistics, the best and second-best at each simulation budget are denoted with **bold** and *italics*, respectively.

Part III

Black-box Neural Posterior Inference for Agent-based Models in the Social Sciences

Chapter 6

A New Generation of Simulation-based Inference Methods[1]

Simulation models in economics and the social sciences – such as agent-based models (ABMs) – are becoming increasingly popular due to the availability of cheap computing power, the greater flexibility they afford modellers, and their ability to reproduce a variety of empirically observed phenomena, such as non-equilibrium behaviour and emergent dynamics within complex social systems. Their widespread employment in real-world modelling and decision-making scenarios is however impeded by the difficulty of performing parameter inference for such models. This difficulty arises due to the fact that these simulation models – as with simulation models in general – lack a tractable likelihood function, which precludes the direct application of standard likelihood-based inference techniques.

A number of approaches to performing statistical inference have been developed in which exact density evaluations are replaced by evaluations of approximate densities or cost functions that are constructed using simulations from the model. Some of these simulation-based inference (SBI) methods have been explored within the ABM community. Perhaps the most prevalent of them is simulated minimum distance (SMD), in which an estimate $\hat{\boldsymbol{\theta}}$ is obtained by minimising some loss function $f(\mathbf{y}, \boldsymbol{\theta})$ between the observed data $\mathbf{y}$ and simulated data $\mathbf{x} \sim p(\cdot \mid \boldsymbol{\theta})$ over some search space

[1]This chapter is based on work appearing in Dyer et al. (2022a) and Dyer et al. (2022b), both of which are joint pieces of work with Patrick Cannon, J. Doyne Farmer, and Sebastian M. Schmon.

$\boldsymbol{\Theta}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} f\left(\mathbf{y}, \boldsymbol{\theta}\right). \tag{6.1}$$

This general class of estimators includes the maximum likelihood estimator – corresponding to choosing, for example, $f\left(\mathbf{y}, \boldsymbol{\theta}\right) = -\log p\left(\mathbf{y} \mid \boldsymbol{\theta}\right)$ or, where the likelihood is intractable, some estimate thereof (see e.g. Diggle and Gratton, 1984; Kukacka and Barunik, 2017). It also includes the Method of Simulated Moments (MSM) (Franke, 2009) – in which f takes the form

$$f(\mathbf{y}, \boldsymbol{\theta}) = \left(g(\mathbf{y}) - \hat{g}_{\boldsymbol{\theta}}\right)' W \left(g(\mathbf{y}) - \hat{g}_{\boldsymbol{\theta}}\right), \tag{6.2}$$

where $g(\mathbf{y})$ denotes a set of moments derived from $\mathbf{y}$, $\hat{g}_{\boldsymbol{\theta}}$ denotes the same set of moments derived from $R \geq 1$ simulations at $\boldsymbol{\theta}$, W is a suitably chosen weight matrix, and $'$ denotes the transpose. As a final example, it includes Indirect Inference (II) (Gourieroux et al., 1993), which follows a similar approach to MSM but replaces the moments with estimated parameters of a tractable auxiliary model. Some further loss functions have been proposed more recently in the context of economic ABM estimation with limited success; see e.g. Platt (2020) for a recent review and benchmarking.

As we have already discussed, a major drawback of SMD and optimisation-based approaches more generally is that only parameter point estimates are produced, while in many contexts it is desirable to obtain meaningful uncertainty quantification regarding appropriate values for $\boldsymbol{\theta}$. As we have also already discussed earlier in this thesis, Bayesian inference is an alternative inferential paradigm which naturally provides this meaningful notion of uncertainty, and has seen some recent but otherwise limited attention within the ABM community (see e.g. Grazzini et al., 2017; Lux, 2018; Platt, 2020, 2021; Lux, 2021).

A well-known investigation into Bayesian estimation methods for ABMs is Grazzini et al. (2017), which explores various means of constructing a surrogate likelihood $\tilde{p}\left(\mathbf{y} \mid \boldsymbol{\theta}\right)$ to account for the intractability of the true likelihood $p\left(\mathbf{y} \mid \boldsymbol{\theta}\right)$. Two immediate drawbacks of the methods Grazzini et al. (2017) discuss are that they (a) are difficult to reconcile with the fact that ABMs are frequently dynamical models generating time-series output, and (b) entail a huge computational burden, since each likelihood evaluation in the employed posterior sampling algorithm requires at least one simulation from the model, and it is typically necessary to make many hundreds of thousands of such evaluations. Recent work by Platt (2021) suggests estimating the model's transition density via a mixture density network. While the approach

offers some improvements through the use of a more flexible density estimator and the incorporation of some temporal dependencies, it suffers from similar drawbacks, assuming time-homogeneity and requiring a computationally expensive estimation step for every single sample from the approximate posterior distribution. In summary, there is a need for parameter inference methods that fulfil two main criteria:

1. they must be simulation-efficient in order to remain applicable to large-scale simulation models such as ABMs;
2. and they must be able to deal with non-homogenous/non-stationary temporal data, both simulated and observed, that exhibit complex dynamics.

To this end, we seek with this chapter to analyse the utility of two classes of parameter inference methods that have seen significant activity within the computational statistics and probabilistic machine learning literature in recent years and that we have reviewed in Section 1.5.4: neural posterior estimation (Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019), and neural density ratio estimation (Thomas et al., 2021; Hermans et al., 2020; Durkan et al., 2020). Such methods have been employed successfully in a variety of applied domains, including high-energy physics (Brehmer et al., 2018), cosmology (Alsing et al., 2019), and neuroscience (Gonçalves et al., 2020). In addition, as we will demonstrate below, they work flexibly with potentially multivariate or even non-Euclidean time-series data, requiring minimal model assumptions, and typically generate more accurate parameter inferences with a significantly reduced simulation budget in comparison to common alternatives, such as those reported in Grazzini et al. (2017).

In summary, our contributions with this chapter are to provide:

1. a motivation for the use of *discriminative* approaches to parameter inference for complex, non-linear, and potentially non-equilibrium models such as ABMs;
2. a systematic benchmarking of these state-of-the-art methods against popular alternatives within the literature on parameter inference for agent-based simulation models in economics and the social sciences;
3. an investigation into the use of novel neural architectures for automatically learning parameter posteriors from high-dimensional sequential data encountered in ABM settings, such as sequences of social networks.

6.1 Motivating black-box, discriminative approaches to parameter inference

Before motivating the methods we consider in this section, we recall and emphasise that a common feature of existing approaches to the task of parameter inference – that is, of constructing the posterior distribution – such as the methods discussed in Sections 1.5.1, 1.5.2, and 1.5.3 is that they are inherently linked to the act of simulating from the ABM and – for some of these approaches – to the act of building an additional approximate *generative* model (sometimes also called an *emulator*), in the sense that an approximation to the likelihood function is constructed from which samples can be generated. We will emphasise below that this contrasts with the new generation of estimation methods described in Section 1.5.4 in which inference is decoupled from the act of simulating and can be performed in a *discriminative* manner, typically resulting in more efficient inferences.

With the above in mind, the methods we investigate in this chapter – neural posterior estimation (NPE) and neural ratio estimation (NRE) – can be motivated by the fact that they are:

1. simulation-efficient alternatives to more traditional approaches to SBI, such as the approximate Bayesian computation (ABC) approaches described in Section 1.5.1;

2. generic SBI methods that treat the simulator as a black-box, thus making minimal assumptions about the structure and output of the simulation model;

3. discriminative approaches to SBI, in the sense that inference does not require a probabilistic model for the data $\mathbf{x}$: instead of *explaining* the mechanisms that create the data, we only need to *distinguish* between realisations that arise from different parameter values.

Simulation efficiency The algorithms described in Sections 1.5.1, 1.5.2, and 1.5.3 share a common pattern. For a fixed parameter $\boldsymbol{\theta}$, *iid* simulations $\mathbf{x}^{(r)} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$, $r = 1, \ldots, R$, are sampled to produce a proxy likelihood $\hat{p}(\mathbf{x} \mid \boldsymbol{\theta})$. Then, to generate n approximate posterior samples via Markov chain Monte Carlo (MCMC), this procedure is performed at n different values for $\boldsymbol{\theta}$, where n must typically be a large number –

often a few hundred thousand, if not orders or magnitude larger – to ensure a low Monte Carlo error. Consequently, at least nR simulations from the ABM are required in total for inference under these algorithms. Since ABMs can be very expensive to simulate, and the act of simulating from the ABM remains the primary bottleneck in Bayesian estimation procedures for ABMs, this simulation demand can quickly become infeasible.

In contrast, the methods for which we advocate here differ by eliminating the need to simulate when sampling from the posterior. Instead, they decouple the act of simulating from the task of constructing the posterior by employing powerful function approximators to learn – in essence – *global* posterior density estimators on the basis of a limited number of simulations from across the parameter space of the ABM; that is, they learn functions $h : \mathcal{Y} \times \boldsymbol{\Theta} \to \mathbb{R}$ which approximate the posterior density $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ across the space of all possible values for $\boldsymbol{\theta}$ and $\mathbf{y}$. This has the potential to significantly reduce the simulation burden associated with approximate Bayesian inference procedures, because the pointwise estimates of $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ can borrow strength from, and share information between, one another; in contrast, the algorithms described in Sections 1.5.1, 1.5.2, and 1.5.3 consider each pointwise evaluation of $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ as standalone density estimation tasks, such that no information can be shared between them. In this way, a large number of approximate posterior samples can be generated from an algorithm trained on what can in practice be a far smaller number of model samples than is required for the algorithms described in Sections 1.5.1, 1.5.2, and 1.5.3.

Finally, the approaches we endorse here are equipped with a further efficiency benefit: *amortisation*. This phrase captures the fact that the global density estimators can be used to generate samples from an approximate posterior $\hat{\pi}(\boldsymbol{\theta} \mid \mathbf{y})$ for any data $\mathbf{y}$ without the need to further simulate from the ABM, which results from the fact that they are trained on the full space of possible values for $\boldsymbol{\theta}$ *and* $\mathbf{y}$. In contrast, applying the methods described in Sections 1.5.1, 1.5.2, and 1.5.3 to a new dataset $\mathbf{y}'$ would entail another nR simulations from the ABM, multiplying the already high simulation costs.

Black-box inference methods We saw in Sections 1.5.2.1 and 1.5.2.2 that many inference approaches come with restrictive assumptions, for instance, stationarity of the simulated and observed time-series. In general, it is useful to dispense with these

assumptions, since they can be difficult to verify and limit the applicability of these methods for arbitrary simulators. Instead, it can be preferable to employ black-box methods that make minimal assumptions about the model and are therefore generically applicable to arbitrary simulators. Doing so enables the modeller to concentrate resources on model design and implementation, rather than on developing bespoke inference algorithms for each new simulator. Furthermore, assumptions such as stationarity are known to be particularly poorly suited to certain economic simulation models such as ABMs, since these models are known and are even designed to produce non-equilibrium dynamics. Employing inference procedures that are able to handle such dynamics is therefore essential to the task of estimating generic ABMs.

Discriminative approaches to parameter inference Discriminative tasks in machine learning are typically simpler than generative tasks, since generative tasks address larger problems than pure discrimination. This is intuitive: for example, it is typically easier for humans and computers alike to distinguish between images of cats and dogs than to generate them. Analogously, it is generally a simpler task to discriminate between complex time-series data than it is to generate such time-series. Many of the approaches described in Sections 1.5.1, 1.5.2, and 1.5.3 adopt a generative approach: they – either explicitly or implicitly – seek to derive an approximation to the simulator's likelihood function using a probabilistic model, and thus seek to model the (probability density function of the) simulation output itself. Formally, this may be understood as learning the (stochastic) map $\boldsymbol{\theta} \mapsto \mathbf{x}$. NPE and NRE, in contrast, do not seek to model the simulation output: instead, they map from instances of the simulation output to certain target values which we will describe in more detail below. This can be formally thought of as learning the stochastic map $\mathbf{x} \mapsto \boldsymbol{\theta}$. Such an approach to parameter estimation therefore embodies a fundamental departure from the approaches described in Sections 1.5.1, 1.5.2, and 1.5.3. It has the potential to be particularly beneficial for ABMs, which are known to be able to produce complex, non-equilibrium dynamics that are especially difficult to model and generate.

6.2 Experiments and demonstrations for tractable examples

In this section, we present experiments in which we compare the ability of NPE and NRE to estimate parameter posterior distributions for economic simulation models with tractable likelihood functions against the non-parametric density estimation method described in Grazzini et al. (2017), which we term KDE and outline in Section 1.5.2.2 as an instance of ABC. We provide details of the neural network architecture and training hyperparameters in Appendix C.4. Throughout, we assume that the density (ratio) estimator includes any embedding network used to learn summary statistics concurrently with the density (ratio) estimate, and thus consider ϕ to contain the parameters of both the estimator and the embedding network (see Section 1.5.5).

6.2.1 Performance metrics

Each example presented in this section will be equipped with a tractable transition density, and therefore a tractable likelihood function. Such models are thus amenable to Bayesian inference via standard means, such as MCMC, to obtain an approximate ground-truth posterior density[2]. As remarked in Section 5.2, we focus on such models in the experiments presented here since it is important to demonstrate new methodology on examples for which a correct answer can be obtained (at least approximately, via methods such as MCMC). The models we use as benchmark experiments below embody a slight departure from the narrative we have presented thus far around the suitability of the endorsed methods for large, complex, and expensive ABMs for this reason, and since the models considered below have either emerged as standard models used in benchmark experiments for ABM calibration methods in the social sciences or have a similar level of complexity to such models (see e.g. Lux, 2018; Platt, 2020, 2021; Lux, 2021, for examples).

To assess the accuracy of the estimated posteriors in this case, we compare the full ground-truth posterior density with the full posterior estimated with each of the implemented simulation-based approaches. This contrasts with previous studies of

[2]While the true posterior density is targeted with such a sampling scheme, the ground-truth obtained in this fashion remains an approximation due to the Monte Carlo error associated with the finite sample size.

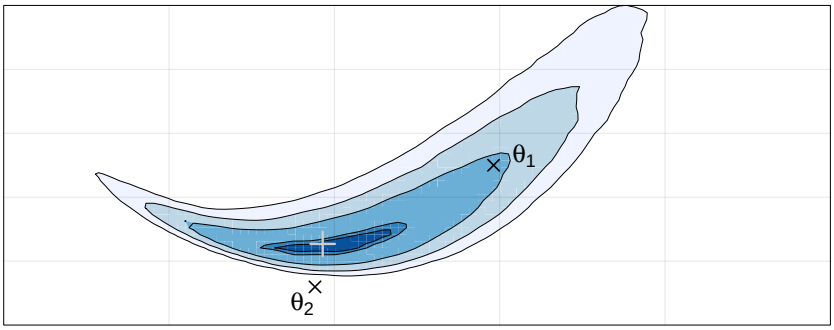


Figure 6.1: Visualisation: why standard Euclidean distances are misleading when gauging the performance of Bayesian inference algorithms. The parameter θ_2 is ostensibly closer to the"true parameter" (grey) in this toy posterior than θ_1. However, θ_1 has higher posterior density, i.e. it is a more credible parameter given the observed data.

Bayesian parameter estimation for ABMs, in which point estimates alone are often used to assess the quality of the tested inference procedures. For example, Platt (2021) uses a prior-weighted Euclidean distance between the estimated posterior mean and generating parameters (that is, the $\boldsymbol{\theta}$ that generated the pseudo-observation $\mathbf{y}$) as a performance metric. However, such metrics provide a very limited and at times misleading view on the outcome of the inference process, because (a) "closesness" is not measured in the geometry of the target distribution, as visualised in Figure 6.1; (b) it is possible that an approximate posterior can produce a posterior mean close to the generating parameter but simultaneously under- or over-estimate the width of the distribution or otherwise yield poor uncertainty quantification; and (c) finite datasets are not guaranteed to yield posterior densities that concentrate on the generating parameter.

To compare the ground-truth and simulation-based posteriors, we compute two integral probability metrics which each correspond to a notion of *dissimilarity* between the ground-truth and estimated posteriors:

Wasserstein distance This is a distance measure derived from optimal transport theory (Kantorovich, 1960) and used widely in various machine learning contexts, e.g. generative adversarial networks (Arjovsky et al., 2017). For some distance ρ_0 on $\boldsymbol{\Theta}$ and two probability measures μ and $\tilde{\mu}$, the p-Wasserstein metric between two sets of

samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{n} \overset{iid}{\sim} \mu$ and $\{\tilde{\boldsymbol{\theta}}^{(i)}\}_{i=1}^{m} \overset{iid}{\sim} \tilde{\mu}$ is computed as

$$\mathcal{W}_p\left(\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{n}, \{\tilde{\boldsymbol{\theta}}^{(j)}\}_{j=1}^{m}\right)^p = \inf_{\gamma \in \Gamma_{n,m}} \sum_{i=1}^{n}\sum_{j=1}^{m} \rho_0(\boldsymbol{\theta}^{(i)}, \tilde{\boldsymbol{\theta}}^{(j)})^p \gamma_{ij} \tag{6.3}$$

where $\Gamma_{n,m}$ is the set of $n \times m$ matrices with non-negative entries, columns summing to m^{-1}, and rows summing to n^{-1}. Throughout, we use the Euclidean distance for ρ_0 and take $p = 1$.

Maximum mean discrepancy (MMD) This metric is once again a metric on probability distributions that draws from the theory of reproducing kernel Hilbert spaces (Gretton et al., 2006, 2012) and is used widely within the machine learning and simulation-based inference community to assess the dissimilarity between probability distributions (Papamakarios et al., 2019; Lueckmann et al., 2021). Here, under a suitable choice of kernel[3] k chosen by the experimenter, the discrepancy between two probability distributions P and Q is taken to be

$$\text{MMD}(P, Q) = \left\| \mathbb{E}_{\boldsymbol{\theta}\sim P}\left[k(\boldsymbol{\theta}, \cdot)\right] - \mathbb{E}_{\boldsymbol{\theta}'\sim Q}\left[k(\boldsymbol{\theta}', \cdot)\right] \right\|_{\mathcal{H}}^2, \tag{6.4}$$

where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) associated with k and $\mathbb{E}_{\boldsymbol{\theta}\sim P}[k(\boldsymbol{\theta}, \cdot)]$ is the so-called *mean embedding* of P via kernel k. When only samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{n} \overset{iid}{\sim} P$ and $\{\tilde{\boldsymbol{\theta}}^{(i)}\}_{i=1}^{m} \overset{iid}{\sim} Q$ are available, an unbiased estimator of this metric can be computed as

$$\widehat{\text{MMD}}\,(P, Q) = \frac{1}{m(m-1)} \sum_{\substack{i,j \\ j\neq i}} k(\tilde{\boldsymbol{\theta}}^{(i)}, \tilde{\boldsymbol{\theta}}^{(j)}) + \frac{1}{n(n-1)} \sum_{\substack{i,j \\ i\neq j}} k(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(j)}) - \frac{2}{nm} \sum_{i,j} k(\boldsymbol{\theta}^{(i)}, \tilde{\boldsymbol{\theta}}^{(j)}). \tag{6.5}$$

Throughout, we use a Gaussian kernel as k,

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp\left(-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2}{2\sigma^2}\right), \tag{6.6}$$

where the scale parameter $\sigma^2 = \text{median}\{\|\tilde{\boldsymbol{\theta}}^{(i)} - \tilde{\boldsymbol{\theta}}^{(j)}\|_2^2\}$, following Briol et al. (2019), and $\{\tilde{\boldsymbol{\theta}}^{(i)}\}_{i=1}^{n}$ are the samples drawn from the ground-truth posteriors.

[3]That is: a symmetric, positive semi-definite function $\boldsymbol{\Theta} \times \boldsymbol{\Theta} \to \mathbb{R}$.

6.2.2 Brock and Hommes (1998)

We consider the simulation model and inference task described in Section 1.7.3. In particular, we consider two experimental setups which we describe further in the following two subsections.

For both NPE and NRE, we use a round-based training approach i.e. sequential neural posterior estimation (SNPE) and sequential neural ratio estimation (SNRE): we train over 10 rounds and generate 1000 simulations in each round. For KDE, we take $R = 1$ and sample from the posterior with Metropolis-Hastings (MH) (see Appendix C.1.1 for further details).

Summarising data As discussed in Section 1.5.5, there exists a number of approaches to representing high-dimensional time-series data $\mathbf{x}$ as a low-dimensional vector $\mathbf{s}(\mathbf{x})$ of summary statistics to facilitate SBI. In the experiments for the Brock & Hommes model, we demonstrate NPE and NRE using summary statistics obtained in two different ways:

1. by summarising them manually as[4] the mean value, variance, maximum, minimum, median, 25th quantile, 75th quantile, and the autocorrelations of the x_t to lags 1, 2, and 3. We denote these summary statistics with $\tilde{\mathbf{s}}$ and refer to them as *naive* or *hand-crafted* summary statistics;

2. by learning them as part of the training process with an embedding network $\mathbf{s}_\varphi$ with trainable parameters φ (see Section 1.5.5). Through the experiments we present below, we will demonstrate that NPE and NRE can be used flexibly with various embedding networks, and that the experimenter is free to choose from the plethora of candidate networks that incorporate useful inductive biases for time-series data (e.g. Wong et al., 2018; Kidger et al., 2020). Below, we refer to summary statistics obtained in this fashion as *learned* summary statistics.

6.2.2.1 Parameter set 1

We take $\beta = 120$ and consider the task of estimating the posterior density $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, where $\boldsymbol{\theta}^* := (g_2^*, b_2^*, g_3^*, b_3^*) = (0.9, 0.2, 0.9, -0.2)$. We use the following uniform priors:

[4] We note however that these are not the only choices available to the experimenter.

$g_2, b_2, g_3 \sim \mathcal{U}(0, 1)$, while $b_3 \sim \mathcal{U}(-1, 0)$.

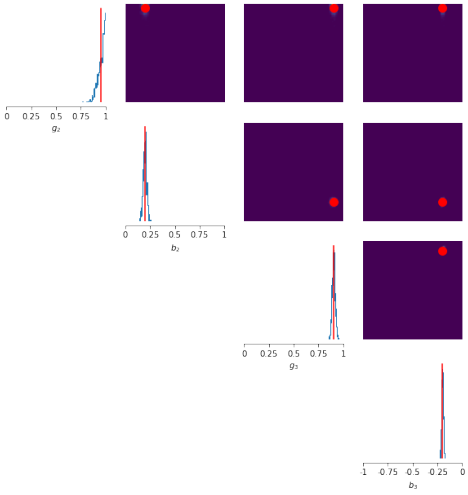
In Figure 6.2, we show the posteriors obtained using different posterior estimation methods: Figure 6.2a shows the approximate ground-truth posterior density obtained with the true likelihood function and MH; Figure 6.2b shows the posterior obtained with KDE and MH; Figures 6.2c and 6.2d show the posterior estimated via SNPE with naive and learned summary statistics, respectively; and Figures 6.3a and 6.3b show the posterior obtained via SNRE and naive and learned summary statistics, respectively, which were also sampled using MH. Here, we use an embedding network consisting of two stacked Elman recurrent units with hidden state of size 32, followed by a single linear layer of size 16. In each figure, the marginals and joint bivariate densities are located on the diagonal and upper diagonal, respectively, while the red lines/dots locate the mean of the true posterior.

We see from the approximate ground-truth in Figure 6.2a that the marginal posteriors are sharply peaked on the generating parameters for b_2, g_3, and b_3, while the ground truth marginal for g_2 is shifted towards higher values. While these features are recovered reasonably well for Figures 6.2c–6.3b (the most notable exception being the posteriors for g_2), we see that they are recovered poorly with KDE. This performance gap is also manifested in the WASSERSTEIN and MMD metrics reported in Table 6.1, where lower values indicate better estimates of the posterior. In summary, KDE both requires a far larger simulation budget to estimate a single posterior, while simultaneously generating worse estimates of that posterior. In contrast, SNPE and SNRE is able to achieve superior estimates of the ground-truth posterior distribution, despite the 10-fold reduction in the simulation budget they have been afforded.

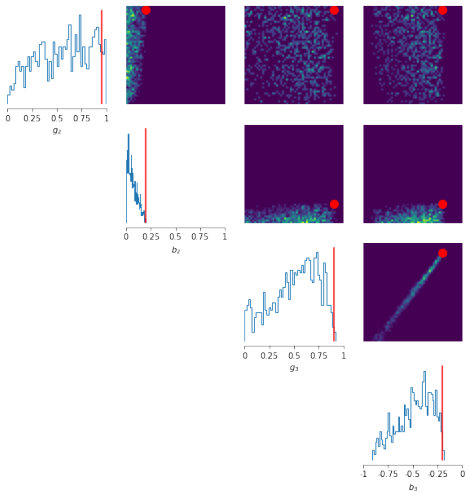
6.2.2.2 Parameter set 2

We now take $\beta = 10$ and consider the task of estimating the posterior density $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, where $\mathbf{y} := (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T) \sim p(\mathbf{x} \mid \boldsymbol{\theta}^*)$, $T = 100$, and $\boldsymbol{\theta}^* := (g_2^*, b_2^*, g_3^*, b_3^*) = (-0.7, -0.4, 0.5, 0.3)$. In this case, we use the following priors: $g_2, b_2 \sim \mathcal{U}(-1, 0)$ and $g_3, b_3 \sim \mathcal{U}(0, 1)$.

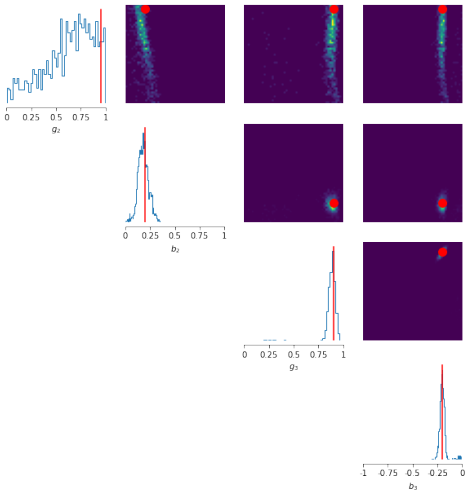
In Figure 6.4, we show the posteriors obtained using different posterior estimation methods: Figure 6.4a shows the approximate ground-truth posterior density obtained with the true likelihood function and MH; Figure 6.4b shows the posterior obtained with KDE and MH; Figures 6.4c and 6.4d show the posterior estimated via SNPE



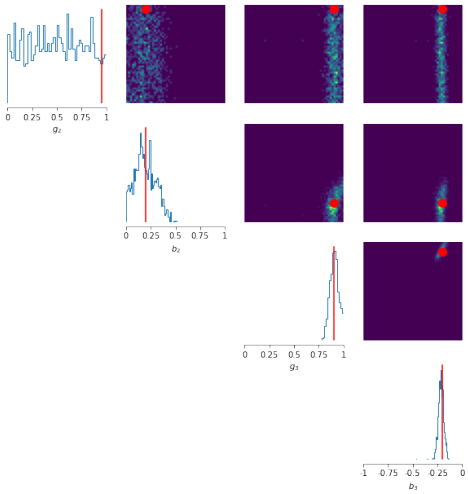
(a) True posterior



(b) Grazzini kernel density estimation (KDE) posterior, 1.5×10^5 simulations

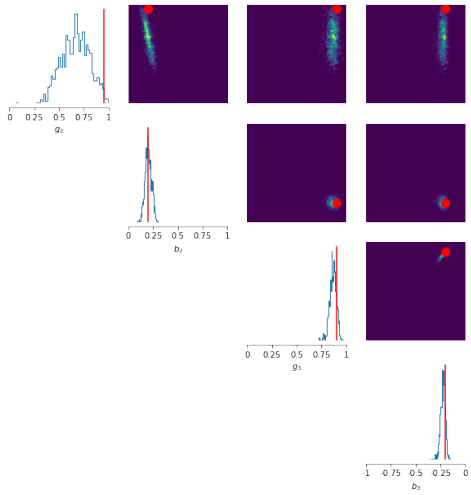


(c) Sequential neural posterior estimation with naive summary statistics, 10^4 simulations

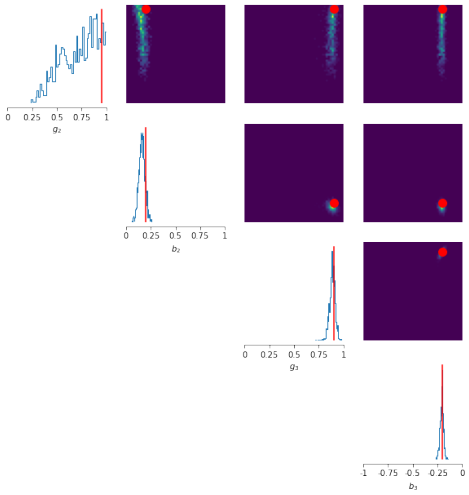


(d) Sequential neural posterior estimation with learned summary statistics, 10^4 simulations

Figure 6.2: (**Brock & Hommes, parameter set 1**) Posteriors obtained with (a) the true likelihood function + MH, (b) KDE + MH, (c) sequential NPE with hand-crafted summary statistics, and (d) sequential NPE with learned summary statistics. The marginal posterior distributions are located on the diagonals, while the bivariate joint distributions for each parameter pair are located on the upper diagonal. Red lines/dots indicate the mean of the true posterior.



(a) Sequential neural ratio estimation with naive summary statistics, 10^4 simulations



(b) Sequential neural ratio estimation with learned summary statistics, 10^4 simulations

Figure 6.3: (**Brock & Hommes, parameter set 1**) Posteriors obtained with (a) sequential density ratio estimation (DRE) + hand-crafted summary statistics + MH, and (b) sequential DRE + learned summary statistics + MH. The marginal posterior distributions are located on the diagonals, while the bivariate joint distributions for each parameter pair are located on the upper diagonal. Red lines/dots indicate the mean of the true posterior.

Parameter set	**Metric**	**Estimation method**				
		KDE	SNPE	SNPE*	SNRE	SNRE*
1	WASSERSTEIN	0.690	0.477	0.336	**0.241**	*0.299*
	MMD	1.015	0.789	*0.552*	**0.451**	0.781
2	WASSERSTEIN	0.304	**0.154**	0.306	*0.164*	0.291
	MMD	0.127	**0.036**	0.133	*0.041*	0.118
Simulation budget		10^5	10^4	10^4	10^4	10^4

Table 6.1: (Brock & Hommes) Discrepancies between the approximate ground-truth posterior and the posteriors estimated with KDE, SNPE, and SNRE. **Bold** and *italics* indicate best and second-best, respectively. For the neural methods, * indicates that the naive hand-crafted summary statistics described in the main text were used, otherwise summary statistics are learned from the simulated data.

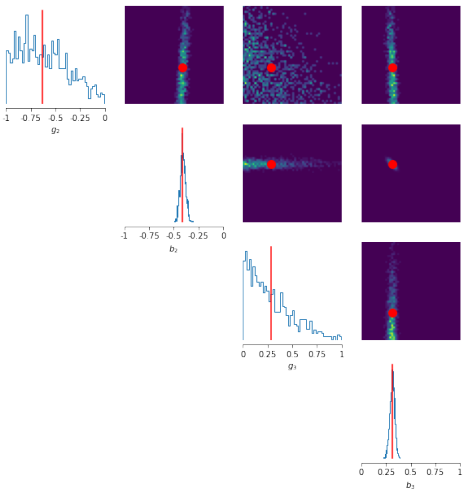
with naive and learned summary statistics, respectively; and Figures 6.5a and 6.5b show the posterior obtained via SNRE and naive and learned summary statistics, respectively, which were also sampled with MH. In this experiment, we now use an embedding network consisting of two stacked gated recurrent units (GRUs) with hidden state of size 32, followed by a single linear layer of size 16. This results in an embedding network with approximately 10,000 trainable parameters.

We see from the approximate ground-truth in Figure 6.4a that the marginal posteriors for b_2 and b_3 remain sharply peaked on the generating parameters, but that the ground truth marginals for g_2 and g_3 are diffuse and sloped. The diffuseness of the posterior with respect to these two dimensions is also visually apparent in the joint density plots in the upper diagonal. Upon inspection of the estimated posteriors, we see that the overall shape is recovered well by SNPE and SNRE with learned summary statistics, but less accurately by KDE and SNPE and SNRE when using the naive hand-crafted summary statistics described in Section 6.2.2. This is once again reflected by the WASSERSTEIN and MMD scores, reported in Table 6.1, in which we see significantly decreased values for SNPE and SNRE with learned summary statistics with respect to the alternatives, despite a 10-fold decrease in the simulation budget they have been afforded. It is nonetheless noteworthy that even with the naive hand-crafted summary statistics, SNPE and SNRE achieve comparable and slightly favourable performance than KDE, again with a 10-fold decrease in the allotted simulation budget.

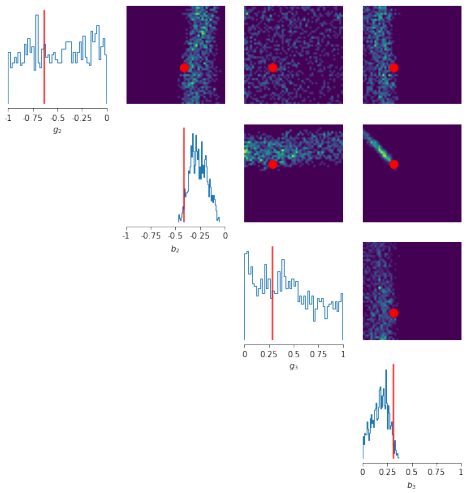
We note that this experiment is identical (i.e. same simulation model, same pseudo-observed dataset, same posterior samples) as the experiment presented in Section 3.2.4. It is therefore also apparent that this improved performance is seen relative to other state-of-the-art instances of ABC and is not limited to the specific case of the KDE method considered in this section.

6.2.3 Multivariate geometric Brownian motion

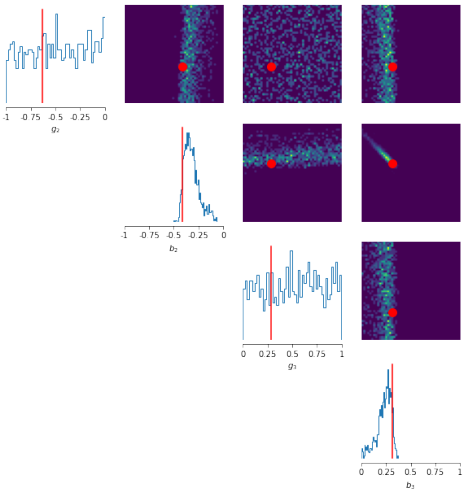
In Section 6.2.2, we demonstrated that NPE and NRE can be used to generate Bayesian parameter posteriors that better match the ground truth posterior than KDE for the univariate Brock & Hommes model, despite the fact that the latter method entailed 10 times as many simulations as the former two methods based on neural networks. In this section, we seek to demonstrate that this remains the case even for models that generate multivariate time-series as output.



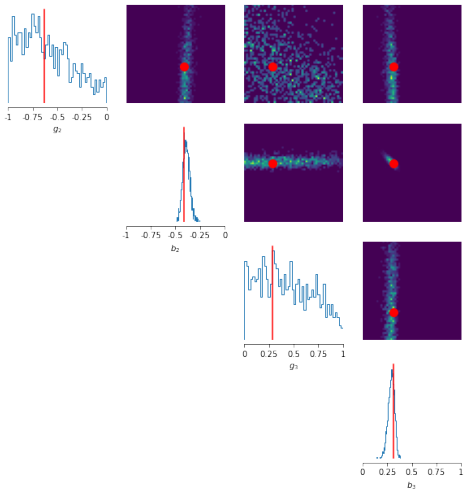
(a) True posterior



(b) Grazzini KDE posterior, 10^5 simulations

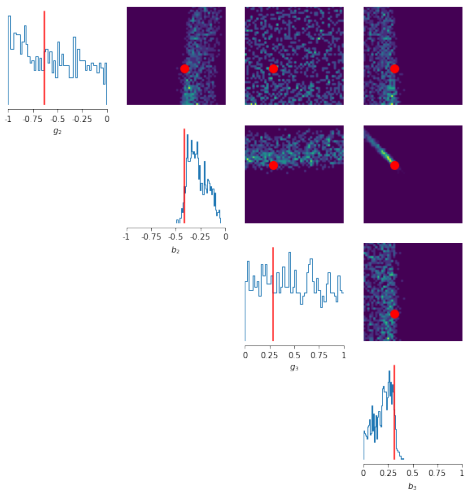


(c) Sequential neural posterior estimation with naive summary statistics, 10^4 simulations

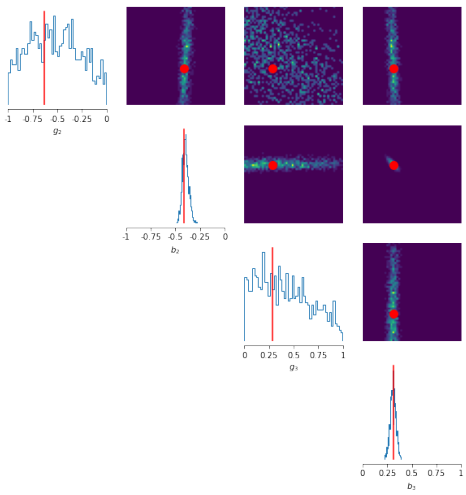


(d) Sequential neural posterior estimation with learned summary statistics, 10^4 simulations

Figure 6.4: (**Brock & Hommes, parameter set 2**) Posteriors obtained with (a) the true likelihood function + MH, (b) the KDE likelihood + MH, (c) sequential NPE + hand-crafted summary statistics, and (d) sequential NPE + learned summary statistics. The marginal posterior distributions are located on the diagonals, while the bivariate joint distributions for each parameter pair are located on the upper diagonal. Red lines/dots indicate the mean of the true posterior.



(a) Sequential neural ratio estimation with naive summary statistics, 10^4 simulations



(b) Sequential neural ratio estimation with learned summary statistics, 10^4 simulations

Figure 6.5: (**Brock & Hommes, parameter set 2**) Posteriors obtained with (a) sequential DRE + hand-crafted summary statistics + MH, and (b) sequential NRE + learned summary statistics + MH. The marginal posterior distributions are located on the diagonals, while the bivariate joint distributions for each parameter pair are located on the upper diagonal. Red lines/dots indicate the mean of the true posterior.

To this end, we consider the multivariate geometric Brownian motion (MVGBM) model described in Section 1.7.2.2. In Figure 6.6a we show the approximate ground truth posterior obtained with MH and the transition density described above, while in Figures 6.6b, 6.6c, and 6.6d we show the posteriors obtained with KDE + MH, NPE with learned summary statistics, and NRE with MH also with learned summary statistics, respectively. The corresponding simulation budgets are 10^6, 10^3, and 10^3, respectively. To learn the summary statistics, we once again use the embedding network described in Section 6.2.2.2 and learn summary statistics and the density (ratio) estimator concurrently.

We see that the approximate ground-truth posteriors are relatively diffuse, with peaks approximately coinciding with the true posterior mean, shown with red lines/dots. The shape and degree of diffuseness is captured accurately by NPE and NRE. In contrast, the posterior obtained with KDE is insufficiently diffuse and biased, and thus a significantly worse estimate of the ground-truth posterior. These observations are corroborated by the corresponding WASSERSTEIN and MMD metrics, reported in Table 6.2 along with the corresponding simulation budgets. In summary, NPE and NRE achieve significantly more accurate posterior estimates here with a 1000-fold decrease in the simulation budget, and are able to flexibly accommodate multivariate

Metric	Estimation method		
	KDE	NPE	NRE
Wasserstein	0.364	**0.099**	*0.107*
MMD	0.137	*0.005*	**0.004**
Simulation budget	10^6	10^3	10^3

Table 6.2: (Multivariate geometric Brownian motion) Discrepancies between the approximate ground-truth posterior and the posteriors estimated with NPE, NRE, and KDE. Smaller values indicate more accurate posteriors; **bold** and *italics* indicate best and second-best, respectively.

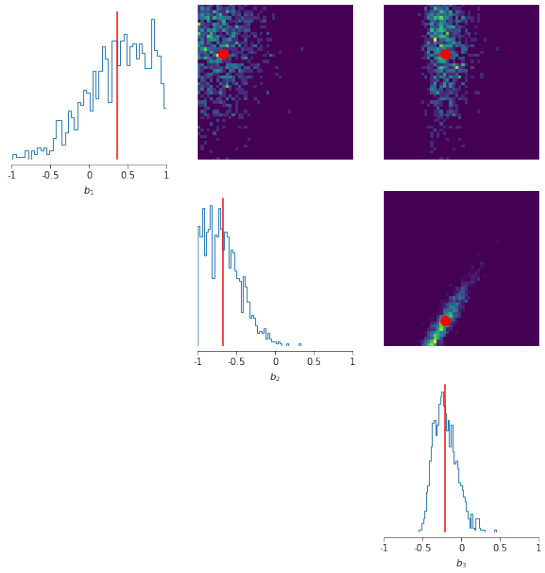
time-series as input for the inference problem.

6.3 Validating approximate Bayesian inference

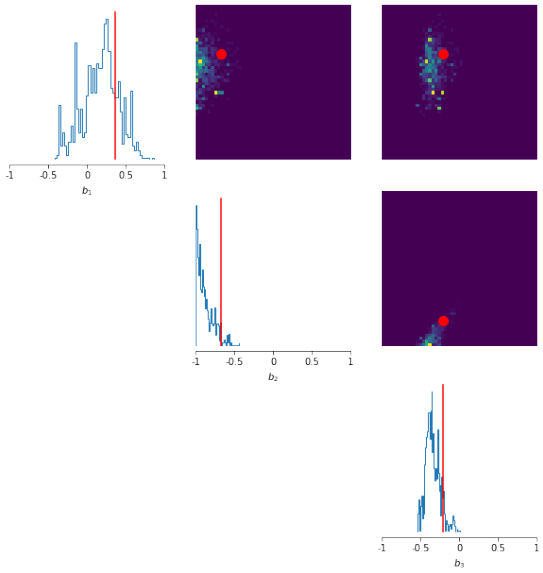
When performing approximate Bayesian inference in practice, it can be difficult to assess whether a posterior generated by any algorithm is in some sense accurate, informative, or meaningful. This raises the question of whether there exists methods for checking the quality of an approximate posterior derived from procedures such as NPE and NRE. In this section, we outline two common approaches to validating approximate Bayesian inference procedures, and discuss how such posterior quality checks are more readily available to the practitioner through the use of NPE and NRE than they are when using more classical posterior inference procedures such as KDE.

6.3.1 Simulation-based calibration

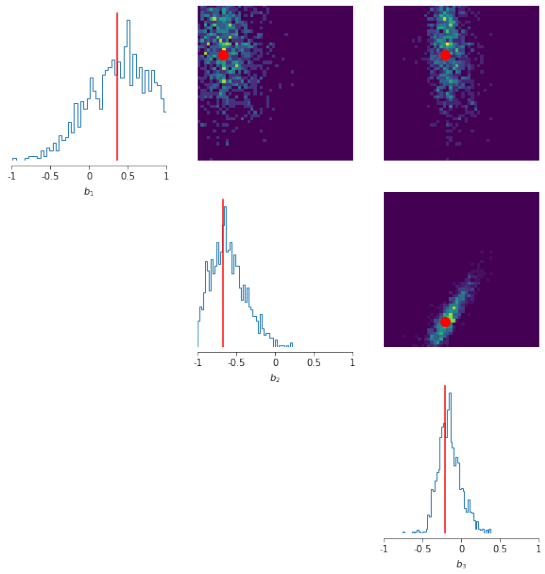
As previously mentioned, a major benefit of the Bayesian inferential paradigm is the fact that uncertainty quantification is a built-in feature captured via the posterior distribution. Its utility relies, however, on the ability to capture the correct degree of diffuseness in the posterior, such that the uncertainty quantification in the recovered posterior is meaningful. This presents a challenge in approximate Bayesian inference, since by definition the experimenter does not know the correct shape of the posterior when exact inference is intractable. Here, we address this issue by outlining a widely



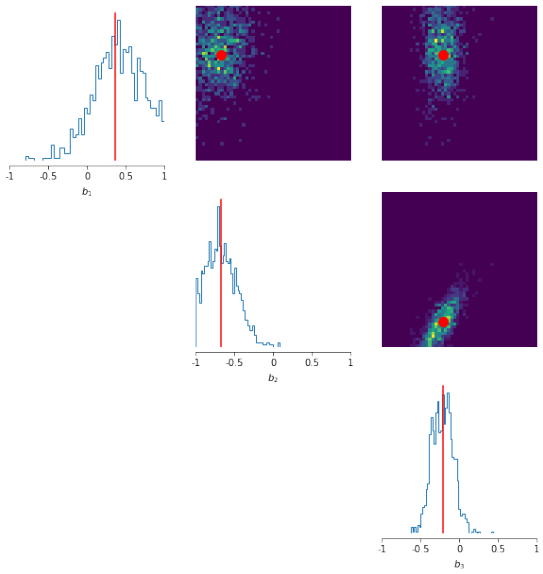
(a) True posterior



(b) Grazzini KDE posterior, 10^6 simulations



(c) Neural posterior estimation with automatically learned summary statistics, 10^3 simulations



(d) Neural posterior estimation with automatically learned summary statistics, 10^3 simulations

Figure 6.6: (**Multivariate geometric Brownian motion**) Posteriors obtained with (a) the true likelihood function + MH, (b) the KDE likelihood + MH, (c) NPE + learned summary statistics, and (d) neural DRE + learned summary statistics + MH. The marginal posterior distributions are located on the diagonals, while the bivariate joint distributions for each parameter pair are located on the upper diagonal. Red lines/dots indicate the mean of the true posterior.

used approach to verifying the accuracy of approximate Bayesian inference pipelines in the absence of any ground-truth posterior densities.

Simulation-based calibration is a general purpose method for validating approximate Bayesian inference pipelines. The core idea is to use the fact that the *data-averaged posterior* should be identical to the prior distribution $\pi(\boldsymbol{\theta})$. That is, a perfect Bayesian inference pipeline should uphold the following equality:

$$\int_{\mathcal{Y}\times\boldsymbol{\Theta}} \pi(\boldsymbol{\theta} \mid \mathbf{y}) p(\mathbf{y} \mid \tilde{\boldsymbol{\theta}}) \pi(\tilde{\boldsymbol{\theta}}) \, \mathrm{d}\tilde{\boldsymbol{\theta}} \, \mathrm{d}\mathbf{y} = \pi(\boldsymbol{\theta}). \tag{6.7}$$

A deviation from this equality signifies some error in the posterior sampling procedure; thus, it has been proposed by Talts et al. (2020) that testing how close our Bayesian workflow is to satisfying this equality is a test of the accuracy of the Bayesian pipeline. This may be performed by repeating the following steps P times:

1. Generate $\tilde{\boldsymbol{\theta}} \sim \pi(\boldsymbol{\theta})$ from the prior distribution
2. Generate $\tilde{\mathbf{y}} \sim p(\mathbf{y} \mid \tilde{\boldsymbol{\theta}})$ from the likelihood function (i.e. the simulator)
3. Generate L uncorrelated posterior samples $\left\{\boldsymbol{\theta}^{(i)}\right\}_{i=1}^{L} \sim \pi(\boldsymbol{\theta} \mid \tilde{\mathbf{y}})$
4. Compute and store the rank statistic $r(\left\{\boldsymbol{\theta}^{(i)}\right\}_{i=1}^{L}, \tilde{\boldsymbol{\theta}}) \in \{0, \ldots, L\}$ for $\tilde{\boldsymbol{\theta}}$ within $\left\{\boldsymbol{\theta}^{(i)}\right\}_{i=1}^{L} \cup \left\{\tilde{\boldsymbol{\theta}}\right\}$

It can be shown (Theorem 1, Talts et al., 2020) that the rank statistics obtained as above should follow a discrete Uniform distribution on $\{0, \ldots, L\}$ for any joint distribution $p(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ for one-dimensional $\boldsymbol{\theta}$, while for d-dimensional $\boldsymbol{\theta}$ each of the d components' rank histograms should be uniform. Inspecting the discrepancy between the true distribution of rank statistics and the desired Uniform distribution thus gives an indication of the accuracy of the Bayesian pipeline; furthermore and conversely, particular patterns of deviation from the desired uniformity are interpretable and provide insight into the specific way in which the obtained posteriors are inaccurate (Talts et al., 2020). Performing this procedure, and inspecting the resultant rank histograms, is referred to as performing simulation-based calibration (SBC).

6.3.1.1 Computational expense of simulation-based calibration

While this procedure is in principle possible for KDE, the computational burden is immense even for cheap simulation models, and becomes completely infeasible for large-scale ABMs. This is because the number of simulations required to generate n samples from the posterior would, in general, be at least[5] nR, such that the total number of simulations required to perform SBC for KDE increases to PnR. Even for a moderately sized SBC task with $P \simeq 10^3$ and the most optimistic $R = 1$, the total number of simulations required can easily reach the order of 10^8, since it is typically necessary to take n to be many hundreds of thousands to obtain a reasonably accurate estimate of the posterior. In fact, larger values of n are likely to be necessary, since ABMs are usually highly parameterised and thus can have large parameter spaces.

In contrast, such an approach to verifying the accuracy of posteriors derived from NPE and NRE is feasible due to the fact that such approaches involve the prior training of a global density (ratio) estimator, obviating any simulations during the inference (posterior sampling) phase. Thus the strength of NPE and NRE derives from not only their enhanced performance and improved simulation-efficiency, but also from the fact that such methods permit accuracy checks such as SBC that are otherwise infeasible for alternative parameter estimation techniques when the simulator is expensive, as is often the case for large-scale ABMs.

6.3.1.2 Example: Franke and Westerhoff (2012)

To further demonstrate the ability of NPE and NRE to recover accurate posterior estimates and informative summary statistics in a highly automated manner, we perform SBC using a posterior and density ratio estimator trained on the Franke & Westerhoff model (Franke and Westerhoff, 2012), which has previously been used in benchmarking experiments for ABM estimation methods (Platt, 2021). In particular, we consider the model version referred to as the "Wealth & Predisposition" model which, similarly to the Brock & Hommes model (see Section 6.2.2), may be written as a system of coupled equations which models the asset price dynamics that result

[5] In practice, this number may be larger due to the necessity of performing trial runs to estimate the parameters of the proposal distribution.

from a heterogeneous system of traders:

$$p_t = p_{t-1} + \mu \left(n^f_{t-1} d^f_{t-1} + n^c_{t-1} d^c_{t-1} \right), \tag{6.8}$$

$$d^f_t = \phi \left(p^* - p_t \right) + \sigma_f \epsilon^f_t, \tag{6.9}$$

$$d^c_t = \chi \left(p_t - p_{t-1} \right) + \sigma_c \epsilon^c_t, \tag{6.10}$$

$$n^f_t = 1 - n^c_t = \frac{1}{1 + \exp\left(-\beta a_{t-1} \right)}, \tag{6.11}$$

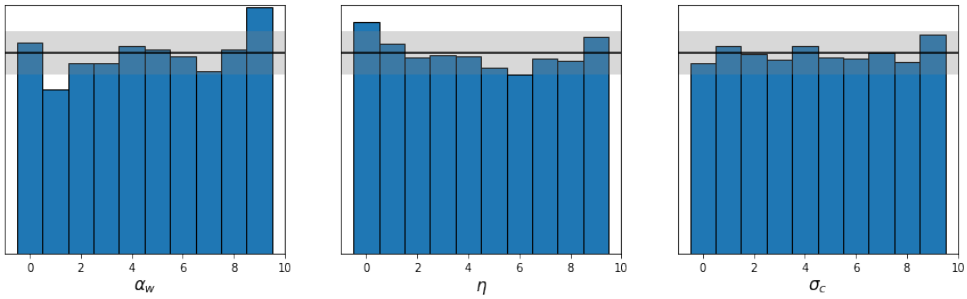
$$a_t = \alpha_w \left(w^f_t - w^c_t \right) + \alpha_0, \tag{6.12}$$

$$w^j_t = \eta w^j_{t-1} + (1 - \eta)\, g^j_t, \quad j \in \{f, c\} \tag{6.13}$$

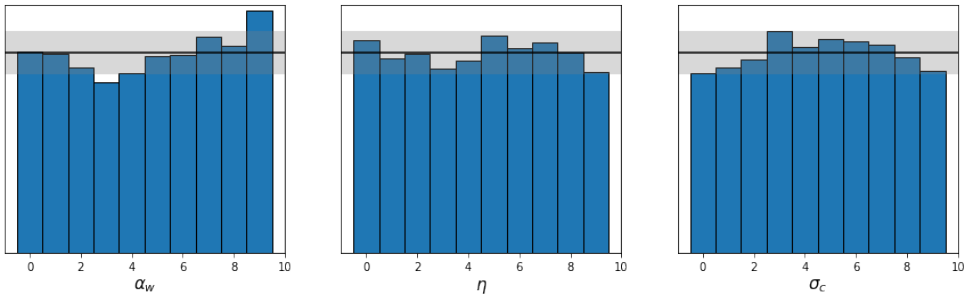
$$g^j_t = \left[e^{p_t} - e^{p_{t-1}} \right] d^j_{t-2}, \quad j \in \{f, c\} \tag{6.14}$$

where the $\epsilon^j_t \sim \mathcal{N}(0, 1)$ are standard normal random variables and $j = \{f, c\}$ labels fundamentalist and chartist traders, respectively. We take as output from the model the time series of log returns $r_t = p_t - p_{t-1}$, and assume the task of training a global, amortised density and density ratio estimator via NPE and NRE, respectively, using 10^4 simulations of length $T = 100$ from the simulator defined by Equations (6.8)–(6.14) with the following parameter settings: $\mu = 0.01$, $\beta = 1$, $\phi = 1$, $\chi = 0.9$, $\alpha_0 = 2.1$, $\sigma_f = 0.752$. The parameter vector for which we seek the posterior and density ratio estimators is taken to be $\boldsymbol{\theta} = (\alpha_w, \eta, \sigma_c)$, and we obtain *iid* samples from the posterior associated with the density ratio estimator via a sampling-importance-resampling scheme described in Appendix C.1.2. (Samples may be drawn *iid* from the NPE posterior by construction.)

We show in Figure 6.7 the rank histograms obtained via the simulation-based calibration procedure in Section 6.3.1 with NPE (Figure 6.7a) and NRE (Figure 6.7b). To construct these histograms, we take $P = 5,000$ and the following uniform priors: $\alpha_w \sim \mathcal{U}(0, 15000)$, $\eta \sim \mathcal{U}(0, 1)$, and $\sigma_c \sim \mathcal{U}(0, 5)$. The black line and gray band denotes the expected height and the expected variation in the heights, respectively, of the bars at this P and with this number of bins. We see that the bars tend to lie within the expected range, although some notable deviations exist: for example, the rank histograms for α_w for both NPE and NRE exhibit a minor bias towards larger rank values, which suggests that the marginal posteriors for α_w in both cases is slightly biased towards lower values on average than the true posteriors (Talts et al., 2020). However, we emphasise that a major benefit of NPE and NRE is that they allow the modeller to make statements of this sort in the first place, whereas the number of simulations required to make such statements in the case of more traditional techniques such as KDE would quickly become prohibitively large.



(a) Simulation-based calibration histogram for neural posterior estimation.



(b) Simulation-based calibration histogram for neural density ratio estimation.

Figure 6.7: (**Franke & Westerhoff**) Rank histograms generated according to the simulation-based calibration procedure in Section 6.3.1 using the trained posterior density estimator (top) and density ratio estimator (bottom).

6.3.2 Posterior predictive checks

Posterior predictive checks (see e.g. Section 6.3, Gelman et al., 1995) are an additional tool for validating approximate Bayesian inference procedures. A posterior predictive check (PPC) captures the notion that, if the inferences that have been drawn about plausible parameter values $\boldsymbol{\theta}$ based on the observed data $\mathbf{y}$ are accurate, then the posterior distribution $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ should assign high probability mass to regions of the parameter space that tend to generate dynamics similar to $\mathbf{y}$. In other words, samples from the posterior predictive distribution,

$$p(\tilde{\mathbf{y}} \mid \mathbf{y}) = \int_{\boldsymbol{\Theta}} p(\tilde{\mathbf{y}} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \mathbf{y}) \mathrm{d}\boldsymbol{\theta}, \tag{6.15}$$

should be in some sense similar to $\mathbf{y}$, and $\mathbf{y}$ should look like a "typical" data point amongst samples from (6.15). Here, $\tilde{\mathbf{y}}$ may denote either future values obtained by simulating forward from the final value in $\mathbf{y}$, or may be taken as hypothetical repetitions of $\mathbf{y}$. Validating approximate Bayesian inference in this way then corresponds simply to repeatedly performing the following set of actions:

1. Sample a parameter from the approximate posterior, $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta} \mid \mathbf{y})$;
2. Simulate a dataset from the model at that parameter value, $\tilde{\mathbf{y}} \sim p(\tilde{\mathbf{y}} \mid \boldsymbol{\theta})$;
3. Store $\tilde{\mathbf{y}}$ for later comparison with $\mathbf{y}$.

Once the $J \geq 1$ posterior predictive samples $\mathcal{S} := \{\tilde{\mathbf{y}}^{(j)}\}_{j=1}^{J}$ have been generated, a comparison between $\mathbf{y}$ and $\mathcal{S}$ may be performed through, for example, a visual inspection by plotting (summary statistics of) $\mathbf{y}$ and the $\tilde{\mathbf{y}}^{(j)}$, or by identifying appropriate test statistics or scoring rules (Section 6.3, Gelman et al., 1995).

6.3.2.1 Computational expense of posterior predictive checks

To simulate $J \geq 1$ realisations $\{\tilde{\mathbf{y}}^{(j)}\}_{j=1}^{J}$ from the posterior predictive distribution, J *iid* samples $\{\tilde{\boldsymbol{\theta}}^{(j)}\}_{j=1}^{J}$ must be generated from the posterior density. The simulation burden for PPCs is therefore the simulation burden associated with construction of the posterior for $\mathbf{y}$ in the first place, in addition to the J further simulations required to perform the PPCs. Since this procedure needs only to be performed for the single dataset $\mathbf{y}$ observed from the real world, the difference in the simulation burden of

PPCs when used the methods we endorse – NPE and NRE – and with more classical methods – such as KDE or other instances of ABC – is likely to be smaller than it is in the case of SBC. However, it remains the case that the simulation burden associated with performing PPCs under NPE and NRE is likely to be orders of magnitude smaller than for more classical approaches to approximate Bayesian parameter inference, due to the greater simulation efficiency of these methods that we explain in Section 6.

6.3.2.2 Example: Macy et al. (2003)

To illustrate the above, we consider an inference task based on the Hopfield model of social dynamics proposed by Macy et al. (2003), which describes the coevolution of opinions and the social network structure, and the emergence of polarisation, in a population of N agents. At each time step $t = 1, \ldots, T$, each agent is equipped with $N-1$ undirected ties to the remaining agents in the population, and the strength and valence of the tie between agents i and j is characterised by $\mathbf{w}_{tij} \in [-1, 1]$. Each agent is also equipped with a state vector $\mathbf{z}_{ti} = (\mathbf{z}_{ti1}, \ldots, \mathbf{z}_{tiK}) \in \{-1, 1\}^K, i = 1, \ldots, N$, which may represent the opinion status of agent i on each of a number $K \geq 1$ of topics at time t. The *social pressure* that agent i experiences on topic k at time t is then modelled as

$$P_{tik} = \frac{1}{N-1} \sum_{j \neq i} \mathbf{w}_{tij} \mathbf{z}_{tik}, \tag{6.16}$$

and i's corresponding propensity to adopt the positive opinion is taken to be

$$\varphi_{tik} = \frac{1}{1 + e^{-\rho \cdot P_{tik}}}, \tag{6.17}$$

where $\rho > 0$ is a free parameter of the model. Agent i then adopts the positive opinion on topic k at time t (i.e. $\mathbf{z}_{(t+1)ik} = 1$) if

$$\varphi_{tik} > 0.5 + \epsilon U_{ti}, \tag{6.18}$$

where $\epsilon \in [0, 1]$ is a further free parameter of the model and $U_{ti} \sim \mathcal{U}(-0.5, 0.5)$. Finally, the ties between agents evolve as

$$\mathbf{w}_{(t+1)ij} = (1 - \lambda)\mathbf{w}_{tij} + \frac{\lambda}{K} \sum_{k=1}^{K} \mathbf{z}_{(t+1)ik} \mathbf{z}_{(t+1)jk}, \tag{6.19}$$

where $\lambda \in [0, 1]$ is a third free parameter of the model.

In this inference task, we deviate from previous experiments and assume the goal of approximating a posterior density for $\boldsymbol{\theta} = (\rho, \epsilon, \lambda)$ having observed the agent-based model *in its entirety*; that is, we take both the agent states $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_T)$ and the inter-agent tie-strengths $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_T)$ over all $T = 25$ time steps – rather than some aggregate time-series derived from these microscopic details – as the output of the model, such that $\mathbf{x} = (\mathbf{z}, \mathbf{w}) \sim p(\cdot \mid \boldsymbol{\theta})$. In this form, the output data $\mathbf{x}$ takes the form of a sequence of weighted, undirected snapshots of an evolving graph, in which the nodes have a set of $K \geq 1$ attributes contained in the state vectors $\mathbf{z}_{ti}$. We assume this more unusual setting in this example as a final illustration of the fact that the methods we investigate in this chapter are immediately able to operate on datasets that may be encountered in the computational social sciences, such as sequences of graphs providing a high-resolution description of the state of a social system.

To be able to derive a parameter posterior directly from this dynamic graph data – rather than from an aggregate time-series that simply summarises this fine, granular dataset – we employ a masked autoregressive flow (Papamakarios et al., 2017) to perform NPE and use as the embedding network a *dynamic graph neural network* (GNN) (see e.g. Seo et al., 2018; Zhou et al., 2020). Dynamic GNNs are neural network architectures that are designed to operate on evolving graph structures – we refer the interested reader to Appendix C.3 for an overview of GNNs and to Appendix C.4 for details on the exact architectures employed. In this way – as with the case of high-dimensional, aggregate, multivariate time-series data discussed in previous sections – NPE and NRE are able to operate on high-dimensional dynamic graph structures by incorporating and leveraging useful inductive biases into their architecture, in this case the inductive biases introduced by dynamic GNNs.

We train this composite network using a budget of 1000 simulations from the ABM described above, and assume prior densities $\rho \sim \mathcal{U}(0, 5)$, $\epsilon \sim \mathcal{U}(0, 1)$, and $\lambda \sim \mathcal{U}(0, 1)$. The inference task is performed for a pseudo-true dataset $\mathbf{y} \sim p(\mathbf{y} \mid \boldsymbol{\theta}^*)$ generated at ground-truth parameter values $\boldsymbol{\theta}^* = (1, 0.8, 0.5)$, and the number of agents in the system is taken to be $N = 50$.

After training the posterior estimator, we perform PPCs by visualising the distribution of two statistics of posterior predictive samples $\tilde{\mathbf{y}} = (\tilde{\mathbf{z}}, \tilde{\mathbf{w}})$:

1. the number of balanced triads in the (signed) adjacency matrix $\boldsymbol{\omega} \in \{-1, 1\}^{N \times N}$ with elements $\boldsymbol{\omega}_{ij} = \text{sign}(\tilde{\mathbf{w}}_{Tij})$. This captures the degree to which the adage

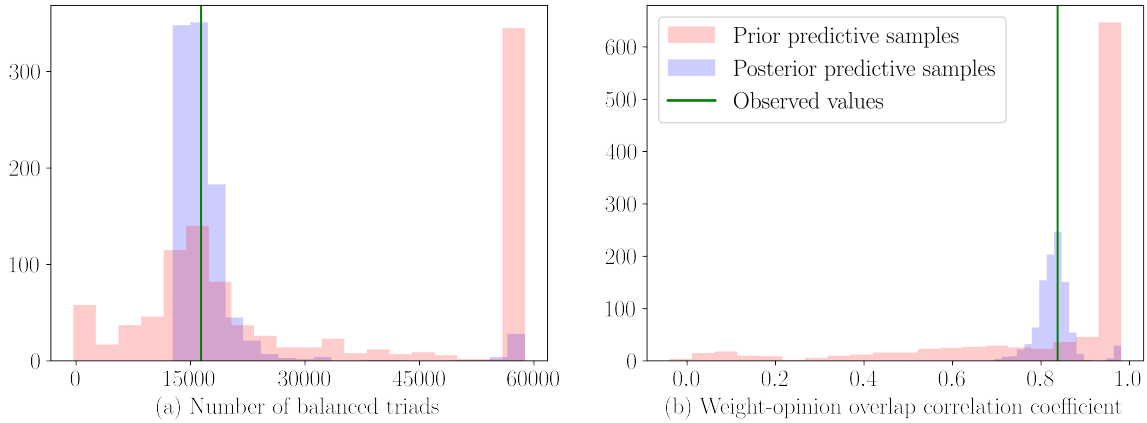


Figure 6.8: (**Social dynamics model**) Distributions of statistics of prior (red) and posterior (blue) predictive samples. Observed statistics are shown with the vertical green line.

"the enemy of my enemy is my friend" is reflected in the final structure of the signed network underlying the simulated social system, and is computed as

$$C = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \boldsymbol{\omega}_{ik} \boldsymbol{\omega}_{kj} \boldsymbol{\omega}_{ki}.$$

2. the linear correlation coefficient between the final tie-strengths $\tilde{\mathbf{w}}_{Tij}$ and the final opinion overlap value,

$$\sum_{k=1}^{K} \tilde{\mathbf{z}}_{Tik} \tilde{\mathbf{z}}_{Tjk}, \tag{6.20}$$

to compare the degree to which the correspondence between the relationship between pairs of agents and the similarity of their opinion profiles matches between the simulated and pseudo-true datasets.

We plot the first and second of these two statistics in the left and right subplots, respectively, of Figure 6.8 for samples from the posterior and prior predictive distributions. We also show the values of these statistics obtained from the pseudo-true dataset $\mathbf{y}$. From this, we see that the observed value of these two statistics is more typical under the posterior predictive samples than it is under the prior predictive samples, suggesting that the posterior estimator has accurately learned how to assign probability mass in appropriate regions of the parameter space.

6.4 Conclusion

In this chapter, we investigated the use of two recently developed approaches to Bayesian parameter estimation for intractable simulation models: neural posterior estimation (NPE) for approximating the posterior density directly, and neural density ratio estimation (DRE) for approximating the likelihood-to-evidence ratio. We motivated the use of these parameter inference methods as simulation-efficient black-box discriminative approaches to Bayesian estimation for complex time-series simulators such as agent-based models in economics and the social sciences, which contrast with existing methods for which at least one of the following is true: (a) they entail an often prohibitively large simulation burden; (b) they make restrictive assumptions about the form of the simulator, for example by ignoring important structural features such as temporal dependencies; and (c) they are inherently generative, and thus assume a task that is typically more difficult than discrimination alone. We argue that this latter point is likely to be particularly pertinent for the case of agent-based models in economics and the social sciences, since they are known to be able to generate complex and stochastic non-equilibrium dynamics that can be difficult to model and generate.

We further reviewed existing alternatives, and benchmarked NPE and neural DRE against one of the most popular of these. In these examples, we see that NPE and neural DRE generally achieve superior recovery of the approximate ground-truth posterior distributions, despite requiring simulation budgets that are orders of magnitude lower than traditional alternatives. In addition, we demonstrated that a verification of the accuracy of the density (ratio) estimators is possible with the introduced methods via simulation-based calibration (SBC), due to the amortisation of the estimators, and furthermore by other well-known approaches such as posterior predictive checks (PPCs). In contrast to NPE and neural ratio estimation (NRE), the former of these – SBC – would necessitate an unreasonably large number of simulations when existing methods for Bayesian estimation of large-scale simulation models in economics and the social sciences are used, for example the methods discussed by Grazzini et al. (2017). For these reasons, we argue that such simulation-efficient black-box Bayesian inference as NPE and NRE may enable economists and social scientists alike to more readily exploit the potential benefits of the agent-based modelling paradigm, due to the dramatic increases in accuracy and correspondingly dramatic decreases in the simulation burden seen with these methods.

Part IV

Epilogue

Chapter 7

Conclusion

The goal of this thesis was to contribute to the literature on likelihood-free, simulation-based inference procedures for dynamic stochastic simulation models. In particular, our goal was to develop methods that are suitable for the kinds of simulation models that are encountered in the social sciences and in social simulation more generally.

To this end, we investigated a variety of approaches to simulation-based Bayesian inference for simulation models that produce sequential data of different kinds. In the first part of this thesis, we framed this problem as one that is made complicated by the intractability of simulation models' likelihood functions. In the second part of this thesis, we considered how an object arising in the theory of controlled differential equations and stochastic analysis – the *signature* of a path – may be employed within different classes of simulation-based inference procedures for dynamic stochastic simulation models. In particular, we showed in Chapter 3 that path signatures may be naturally and successfully embedded into approximate Bayesian computation pipelines, and discussed the theoretical properties of such an approach, along with its behaviour in misspecified settings. We further discussed in Chapters 4 and 5 how path signatures may be employed in density ratio estimation methods, in which the likelihood-to-evidence ratio in Bayes's theorem is approximated with probabilistic classifiers. In particular, we considered how such an approach may be especially useful for expensive time-series simulators, of which there are many examples in the social sciences (e.g. large-scale agent-based models).

In the third part of the thesis, we investigated in more detail how two recently developed approaches to simulation-based inference that have emerged from the proba-

bilistic machine learning community – neural posterior estimation and neural density ratio estimation – offer a promising approach to simulation-based Bayesian parameter inference for agent-based models, which are of significant interest in economics and the social sciences more broadly. We motivated the use of these methods as flexible, simulation-efficient, and accurate approaches to approximate Bayesian parameter inference that simultaneously address many of the key challenges faced when calibrating parameter for these models, and demonstrated with experiments how they may be straightforwardly and automatically applied to not only multivariate time-series data, but also to more exotic scenarios such as models that simulate evolving graphs. Such an approach may prove useful in social simulation settings, such as when modelling evolving social networks or occupational mobility networks in labour markets (del Rio-Chanona et al., 2021).

We hope that the work presented in this thesis serves to inspire further work in the development of inference techniques, and methodology more generally, that make social simulation an informative and robust tool for use by decision- and policy-makers. With this in mind, we conclude by considering some limitations and challenges faced in simulation-based Bayesian parameter inference, and how these challenges relate to the methods we discuss in the previous chapters. We also consider some broader challenges in the development of robust, data-driven simulation models in the social sciences.

Choice of neural network architecture Previously, we have discussed how the methods we endorse in Chapter 6 are able to flexibly accommodate different kinds of data and simulators by choosing a network architecture that incorporates appropriate inductive biases – network architectures that reflect the specific structure of the simulator, or the data generated by the simulator. While this offers considerable flexibility to the practitioner, and can result in powerful posterior estimators and eliminate the need to manually construct summary statistics of the data, it nonetheless leaves open the precise choice of architecture and network design. It may be argued, therefore, that a degree of arbitrariness remains, and that the problem of designing an appropriate set of summary statistics that describe the data has simply been replaced with the similarly difficult task of designing an appropriate neural architecture. We submit, however, that (a) the flexibility and (b) the reduction in the required simulation burden that results from the use of NPE and NRE outweighs this drawback:

(a) there exists a vast and rapidly growing literature in which the performance of different neural network architectures is thoroughly tested and compared, which will provide the practitioner with expert guidance on appropriate network architecture choices. Furthermore, there is significant interest and progress being made in developing neural network architectures that are robust to complex and messy data settings – such as neural controlled differential equations (Kidger et al., 2020) for irregularly sampled data with missing values, or temporal graph networks (Rossi et al., 2020) for streams of edges in social networks – which are of great practical relevance to economists and computational social scientists seeking to use such data to calibrate agent-based models (ABMs);

(b) in all methods, experimentation with hyperparameters will be required (e.g. network architecture for NPE and NRE; choice of distance function in approximate Bayesian computation (ABC); choice of kernel in KDE etc.). The key point is that, when more traditional methods such as KDE are used to approximate the parameter posterior, such experiments will entail a number of simulations that is orders of magnitude larger than the number of simulations that will likely be required for corresponding experiments with NPE and NRE. When the cost of simulating is large – as it typically will be for, for example, large-scale ABMs in economics and the social sciences – this will allow more rapid iterations and an ability to appropriately asses the posteriors/posterior estimators, for example through the methods described in Section 6.3.

Expense of training The methods we endorse in Chapter 6 – NPE and NRE – have been seen to entail a simulation burden that is typically orders of magnitude smaller than more traditional parameter inference methods for ABMs. However, NPE and NRE entail an additional training time that is not shared by some other more traditional techniques. In general, we submit that, for large-scale ABMs, this training time will likely often be a small proportion of the time spent simulating; this is particularly the case when the practitioner has access to specialised hardware such as GPUs, which are by now taken for granted by computational scientists and access to which is relatively straightforward through e.g. cloud computing platforms.

Robustness to misspecification Throughout the majority of this thesis (i.e. everywhere, with the exception of Section 3.1.1.2), we have implicitly assumed that the

simulators we considered are accurate descriptions of reality. More precisely, we have assumed throughout that the simulator is a *well-specified* model, meaning that there exists some $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ such that $p(\cdot \mid \boldsymbol{\theta}) = p_*$, where p_* is the density from which the observed data $\mathbf{y}$ was drawn, $\mathbf{y} \sim p_*$. In practice, however, models will be definition be misspecified – every model necessarily omits some aspect of reality – such that for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ we have $p(\cdot \mid \boldsymbol{\theta}) \neq p_*$. For the modeller, it is then a question of how poorly specified the model is, and whether any inference procedure employed for e.g. parameter calibration can appropriately handle this misspecification. This is an active area of research for inference procedures in general (see e.g. Miller and Dunson, 2018; Knoblauch et al., 2019; Schmon et al., 2020), and is a problem relevant to all inference procedures (i.e. not only NPE and NRE). While there has been quite significant investigation into this problem within ABC (an exhaustive list is not possible here, but we refer the reader to e.g. Frazier et al., 2020b,a; Schmon et al., 2020), there is currently limited work on assessing or improving the robustness of NPE and NRE to model misspecification – see Cannon et al. (2022) for a recent example – and further work is needed to understand how well such approaches generalise to misspecified settings.

Differentiable agent-based models Much of modern machine learning – specifically, deep learning – relies on the differentiability of function approximators, such that gradient-based optimisation procedures can be employed to calibrate parameter values for optimal performance. This raises the question of whether parameterised mechanistic models, such as ABMs, can also benefit from such approaches to parameter estimation. The reason that this does not currently translate to ABMs is that ABMs are usually not differentiable, in that the operations comprising the forward simulation of ABMs are not typically differentiable operations. It is therefore not immediately possible to exploit modern autodifferentiation frameworks or gradient-based optimisation procedures to tune ABM parameters, since we cannot usually differentiate through the internal operations of ABMs. However, an interesting line of work would be to develop automatic procedures to make arbitrary ABMs differentiable, for example by replacing non-differentiable operations with differentiable approximations to the original operation. Some recent work in this direction includes Chopra et al. (2022), in which epidemiological ABMs are the primary object of interest. Developments in this area would constitute useful and interesting contributions to the literature on agent-based modelling methodology.

Model synthesis Our focus throughout this chapter, and indeed this thesis more generally, has been on the problem of inferring parameters for (agent-based) simulation models. However, there exist other learning tasks relevant to constructing useful mechanistic models that can be used in real-world scenarios. One such task is the problem of model synthesis; that is, learning the operations themselves that comprise the simulation. This is the subject of *inverse generative social science*, a research area in which there has already been a number of interesting developments (see e.g. Vu et al., 2019, 2022)[1]. For example, Greig and Arranz (2021) learn symbolic models of flocking behaviour and of opinion dynamics using a genetic algorithm. An interesting direction for future work in this area would be to develop principled methods for incorporating prior knowledge about behavioural rules into the learning procedures, in order to reflect knowledge already derived from e.g. psychological or behavioural experiments.

Sequential inference schemes The typical inference problem set-up in the neural simulation-based inference (SBI) literature is that all data is observed beforehand in an offline manner, and a single parameter posterior is derived based on this complete observation. However, in many practically relevant settings – including the setting of ABMs – data may arrive sequentially or in an online manner, and in such cases it would be desirable to have neural SBI methods at one's disposal that can account for data that continues to arrive at later times. A useful research direction would therefore be to adapt existing neural SBI techniques to these settings, allowing the experimenter to easily update inferences on the basis of new data.

Applying these methods to, and testing these methods on, larger and more complex models used in the social sciences The focus of this thesis has been on developing parameter inference methodology that is applicable to stochastic simulation models that arise in the social sciences. To this end, we have attempted to distill such models to their primary features that make this task difficult. Aside from the general difficulty of working with intractable likelihood functions faced in many settings involving simulation models, we identified key features of simulation models in the social sciences that can make this task challenging:

[1]This is conceptually a very closely related problem to that of parameter inference – which has been the main focus of this thesis – in the sense that, in both cases, one's goal is to search over a space of possible models for ones that are consistent with the observed data.

1. they generate non-*iid*, time-series data in many cases;
2. these time-series data may be complex in different ways – for example, they may exhibit complex dynamics, or may be multivariate, or consist of irregularly spaced observations, or may even consist of graph-valued observations when they are used to model complex systems of interacting entities; and
3. they can be expensive to simulate, due to the number of interacting entities they often must simulate.

The models on which we test the methods proposed and discussed in this thesis, while significantly smaller and simpler than some models that may arise in real-world applications of simulation modelling in the social sciences, were selected on the basis that they possess at least one of these features, and therefore share some of the properties that are expected from larger and more complex simulation models that may appear elsewhere in the social sciences. A good performance of the methods we discuss on such examples can therefore give us some confidence that this good performance would extend to larger simulation models in the social sciences that share these features.

Through benchmarking experiments on these simpler models, we showed that these methods can outperform existing alternatives – often by a significant margin – on these comparatively simple models. It would be reasonable to expect, in light of this fact, that these new methods would continue to perform at least as well as existing alternatives when applied to larger and more complex simulation models: if existing alternatives cannot perform well on simple examples, it is difficult to expect that they will not perform poorly when the complexity of the problem increases.

Additionally, there are existing instances of interesting and useful simulation models in economics and the social sciences that are not significantly different in structure and number of parameters to the examples considered in this thesis. For example, del Rio-Chanona et al. (2021) provide a model of occupational mobility consisting of workers undergoing a random walk on a network in which nodes are occupations and edge weights represent the probability with which workers transition from one occupation to another. After extracting the values of certain model parameters directly from data on the real-world counterparts of those quantities – for example, the conditional probability $\mathbb{P}(\{\text{the worker transitions to occupation } j\} \mid \{\text{the worker transitions to a different occupation}\})$ is taken directly from a real-world

dataset on real worker occupation transitions – the authors are left with four free parameters to calibrate, which is the same dimensionality as the Brock & Hommmes experiment we consider in Section 6.2.2.2, for example. Using this calibrated model, the authors were able to closely mimic and reproduce a well-known feature of labour markets – the empirical Beveridge curve, which describes the relationship between the unemployment rate and job vacancy rate. We furthermore note that the model proposed in del Rio-Chanona et al. (2021) is an example of a real-world economics simulation model which may be taken to generate a sequence of graphs as output – for example, a graph representing the number of transitions between occupations in the occupation network – and therefore for which the graph-based methods proposed in Sections 3.2.6 and 6.3.2.2 will be suitable, which demonstrates the suitability of the methods proposed and discussed in this thesis to real-world simulation models in economics and the social sciences.

However, obtaining concrete, empirical evidence of the good performance of these new methods on larger, more complex models would nonetheless be very informative[2]. An example of such a model is Baptista et al. (2016), which is an agent-based model of the UK housing market developed by the Bank of England and academics at the Institute for New Economic Thinking, Oxford. The model represents the housing market with three types of agents – each agent is either a household, a mortgage lender, or a central bank – and the model proceeds by having the agents choose from a discrete set of possible decisions and actions at each time step, for example the decision by a household to purchase a property outright if the financial wealth of the household is more than twice the price of the property. The model generates a multivariate time-series consisting of such quantities as the house price index over time, and is large and far more highly parameterised than the exampled considered in this thesis, containing approximately 10,000 households and with approximately 40 free parameters to calibrate.

To begin to address the question of the extent to which the methods discussed in this thesis would be capable of tackling a calibration problem such as the one posed by this housing market model, it is worthwhile to note that some previous attempts to calibrate the model have been limited in that they have attempted to calibrate

[2]The difficulty here – as discussed previously in this thesis – is that it can be hard to assess whether a Bayesian inference pipeline is producing a meaningful or "correct" posterior density in such cases, since there will usually be no ground-truth posterior density to compare against (although as discussed in the previous chapter, posterior predictive checks provide us with some incomplete indication that the approximate posterior is meaningful).

to only a univariate time series output from the model, due to the limitations of the employed calibration procedure (Platt, 2020). In contrast, the methods that we develop, propose, and endorse in this thesis would naturally and immediately be able to handle multivariate time-series generated as output from the simulation model. Additionally, due to the sheer quantity of agents present in this model, some of the methods we have proposed and investigated in this thesis – which we have demonstrated to entail a significantly reduced simulation burden in comparison to competing methods – will make the process of calibrating the model more computationally feasible. It is, however, difficult to make any claims about the performance of these methods in highly parameterised settings without experimentation with such examples, and a useful direction for future research would be to devise simulation models that are highly parameterised but that are nonetheless sufficiently simple as to permit an approximate ground-truth posterior density. This would enable us to properly test the performance of these methods in parameter calibration tasks that involve high-dimensional parameter spaces.

Appendix A

Background on rough paths

In this section, we provide some basic definitions and results in the theory of rough paths that are used or discussed in the main text. Throughout this section, V will be a Banach space and $\Delta_{[0,T]} := \{(s,t) \in [0,T]^2 : 0 \leq s \leq t \leq T\}$.

Definition 4 (Tensor algebra)**.** *The* truncated tensor algebra *at integer degree n over V*

$$T^{(n)}(V) := \left\{ s \in \prod_{k=0}^{n} V^{\otimes k} := \mathbb{R} \oplus V \oplus (V \otimes V) \oplus \cdots \oplus V^{\otimes n} \;\middle|\; s_0 = 1 \right\},$$

where s_0 indicates the first element of $s \in T^{(n)}(V)$. The extended tensor algebra *$T((V))$ is the infinite sequence $\prod_{n\geq 0} V^{\otimes n}$.*

With this definition in place, we can now define a multiplicative functional.

Definition 5 (Multiplicative functional, Definition 3.1 of Lyons et al. (2007))**.** *Let $n \geq 1$ be an integer, and $X : \Delta_{[0,T]} \to T^{(n)}(V)$ be a continuous map. For each $(s,t) \in \Delta_{[0,T]}$, denote by*

$$X_{s,t} := (X^0_{s,t}, X^1_{s,t}, \ldots, X^n_{s,t}) \in \prod_{k=0}^{n} V^{\otimes k}$$

the image of (s,t) under X. If $X^0_{s,t} = 1$ $\forall (s,t) \in \Delta_{[0,T]}$ and

$$X_{s,u} = X_{s,t} \otimes X_{t,u} \quad \forall s,t,u \in [0,T] \text{ s.t. } s \leq t \leq u,$$

then X is called a multiplicative functional *of degree n in V.*

Remark 6. *The path signature for a bounded variation path $X : [0,T] \to V$ truncated to some finite degree M is an element of the truncated tensor algebra at degree M over V, and is a multiplicative functional as a result of Chen's identity (Chen, 1958), giving that*

$$S_{s,u}^{\leq M} = S_{s,t}^{\leq M} \otimes S_{t,u}^{\leq M} \quad \forall s,t,u \in [0,T] \text{ s.t. } s \leq t \leq u,$$

where $S_{s,t}^{\leq M}$ denotes the collection of the first M tensors in the signature integrated over $(s,t) \in \Delta_{[0,T]}$.

Definition 6 (Rough path, Definition 3.11 of Lyons et al. (2007))**.** *Let $p \geq 1$ be a real number. A p-rough path in V is a multiplicative functional of degree $\lfloor p \rfloor$ in V with finite p-variation. The space of such paths is denoted with $\Omega_p(V)$.*

The behaviour of rough paths may be described through the notion of a *control*:

Definition 7 (Control functions, Definition 1.9 of Lyons et al. (2007))**.** *A* control function*, or simply* control*, on $[0,T]$ is a continuous non-negative function ω on $\Delta_{[0,T]}$ which is super-additive in the following sense:*

$$\omega(s,t) + \omega(t,u) \leq \omega(s,u) \quad \forall s,t,u \in [0,T] \text{ s.t. } s \leq t \leq u$$

and $\omega(t,t) = 0 \ \forall t \in [0,T]$. If, for a continuous path $X : [0,T] \to V$ and for all $(s,t) \in \Delta_{[0,T]}$, $\|X\|_{p-\mathrm{var},[s,t]} \leq \omega(s,t)^{1/p}$ for some $p \geq 1$, then we say that the p-variation of X is controlled by ω.

A broad and useful class of rough paths – geometric p-rough paths – may be expressed as a limit of bounded variation paths in the following metric:

Definition 8 (The p-variation metric)**.** *Let $p \geq 1$ be a real number, and $C_{0,p}(\Delta_{[0,T]}, T^{(\lfloor p \rfloor)}(V))$ be the space of all continuous functions from $\Delta_{[0,T]}$ to the truncated tensor algebra $T^{(\lfloor p \rfloor)}(V)$ with finite p-variation. The p-variation metric between $X, Y \in C_{0,p}(\Delta_{[0,T]}, T^{(\lfloor p \rfloor)}(V))$ is defined as*

$$d_p(X,Y) = \max_{1 \leq i \leq \lfloor p \rfloor} \sup_{\zeta(0,T)} \left(\left\| X^i_{t_{l-1},t_l} - Y^i_{t_{l-1},t_l} \right\|^{\frac{p}{i}} \right)^{1/p},$$

where the supremum is taken over finite partitions $\zeta(0,T)$ of $[0,T]$.

Equipped with this metric, geometric p-rough paths are defined in the following way:

Definition 9 (Geometric p-rough path, Definition 3.13 of Lyons et al. (2007))**.** *Let $p \geq 1$ be a real number. A* geometric p*-rough path in V is a p-rough path that can be expressed as a limit of* 1*-rough paths in the p-variation metric. The space of such paths is often denoted $G\Omega_p(V)$, and $G\Omega_p(V) \subset \Omega_p(V)$.*

The space $G\Omega_p(V)$ of geometric p-rough paths is therefore the closure of $BV([0,T],V)$ in $(\Omega_p(V), d_p)$ and encompasses a broad range of paths, e.g. fractional Brownian motion with Hurst parameter $> 1/4$ and continuous-time Markov processes. The following two results show that the signatures of such (geometric) p-rough paths are well-defined and continuous in an appropriate topology.

Theorem 4 (Extension Theorem, Theorem 3.7 in Lyons et al. (2007))**.** *Let $p \geq 1$ be a real number, $n \geq \lfloor p \rfloor$ an integer, and $X : \Delta_{[0,T]} \to T^{(n)}(V)$ a multiplicative functional with finite p-variation controlled by ω. Then there exists a unique extension of X to a multiplicative functional $\Delta_{[0,T]} \to T((V))$ which possesses finite p-variation.*

Theorem 5 (Continuity of the Extension Map, Theorem 3.10 in Lyons et al. (2007))**.** *Let X, Y be two multiplicative functionals in $T^{(n)}(V)$ of finite p-variation with $n \geq \lfloor p \rfloor$ an integer, controlled by ω. Suppose that for some $\epsilon \in (0,1)$*

$$\left\| X^i_{s,t} - Y^i_{s,t} \right\| \leq \epsilon \frac{\omega(s,t)^{\frac{i}{p}}}{\beta\left(\frac{i}{p}\right)!} \tag{A.1}$$

for $i = 1, \ldots, n$ and for all $(s,t) \in \Delta[0,T]$. If

$$\beta \geq 2p^2 \left(1 + \sum_{r=3}^{\infty} \left(\frac{2}{r-2} \right)^{\frac{\lfloor p \rfloor + 1}{p}} \right),$$

then (A.1) holds for all i.

This leads us to the definition of the signature of a geometric p-rough path:

Definition 10 (The signature of a geometric p-rough path)**.** *The signature of a geometric p-rough path $X \in G\Omega_p(V)$ with p-variation controlled by some control ω is defined to be the unique extension of X to a multiplicative functional in $T((V))$ under the Extension Theorem, Theorem 4.*

Appendix B

Deep Signature Transforms

We summarise the components of deep signature transforms, following (Kidger et al., 2019).

Stream-preserving feature map

The learnable, stream-preserving neural network $\Phi^{\varphi} : \mathbb{R}^{d \times m} \to \mathbb{R}^{e}$ for some $m \in \mathbb{N}$ operates on the original stream $\mathbf{x}$ as

$$\Phi(\mathbf{x}) = (\Phi_1, \ldots, \Phi_{n-m+1}),$$

where $\Phi_k = \Phi^{\varphi}(x_k, \ldots, x_{k+m}; \Phi_{k-1})$ and $\Phi_0 = 0$. This general structure can take the form of a one-dimensional convolutional layer, a feedforward network, or recurrent network.

Lift operation

The learnable feature map obtained from the stream-preserving neural network augments the existing stream with additional channels. Its operation is described as "stream-preserving" since it does not destroy the stream-like nature of the data. The signature transform, on the other hand, operates on streams to produce an infinite set of features with no inherent stream-like properties. Direct application of the signature transform will thus prohibit its further application.

In general, however, we may wish to apply the signature transform repeatedly. This motivates the inclusion of a lift operation between the learnable, stream-preserving network and the signature transformation. Letting $\mathcal{S}(\mathbb{R}^d)$ be the space of sequences in $\mathbb{R}^d$, a lift operation $\ell : \mathcal{S}(\mathbb{R}^d) \to \mathcal{S}(\mathcal{S}(\mathbb{R}^e))$ for some $e \in \mathbb{N}$ maps a stream into the space of streams of streams. Applying the signature transform element-wise to the lifted stream therefore yields a stream of signatures,

$$\mathrm{Sig}_N(\ell(\mathbf{x})) := (\mathrm{Sig}_N(\ell_1(\mathbf{x})), \ldots, \mathrm{Sig}_N(\ell_v(\mathbf{x}))) \in \mathcal{S}(\mathbb{R}^{(e^{N+1}-1)/(e-1)}),$$

which is amenable to further signature-based analysis (because the output is a stream). Examples of a lift operation include expanding windows $\ell(\mathbf{x}) = (\tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3, \ldots, \tilde{\mathbf{x}}_n)$ where $\tilde{\mathbf{x}}_i = (\mathbf{x}_1, \ldots, \mathbf{x}_i)$, or sliding windows with window length p, in which case $\ell(\mathbf{x}) = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_n)$ and $\tilde{\mathbf{x}}_i = (\mathbf{x}_i, \ldots, \mathbf{x}_{i+p})$.

Neural-lift-signature block

A stream-preserving neural network can be combined with a lift-signature operation to create a neural-lift-signature block

$$B_N^{\varphi}(\mathbf{x}) = (\mathrm{Sig}_N \circ \ell \circ \Phi^{\varphi})(\mathbf{x}).$$

This composite operation may or may not be stream-preserving. In particular, a neural-lift-signature block is not stream-preserving if we take $\ell(\mathbf{x}) := \mathbf{x}$ for that block.

Deep Signature Transforms

Let $\mathcal{X}$ be some set and $f^{\varphi} : \mathcal{S}(\mathbb{R}^c) \to \mathcal{X}$ be a neural network with trainable parameters φ. A deep signature transform $\sigma(\mathbf{x})$, illustrated in Figure 4.1, is a mapping from $\mathcal{S}(\mathbb{R}^d)$ to $\mathcal{X}$ defined as any sequence of k neural-lift-signature blocks followed by an optional final neural network $f^{\varphi_{k+1}}$, i.e.

$$\sigma^{\varphi}(\mathbf{x}) = \left(f^{\varphi_{k+1}} \circ B_{N_k}^{\varphi_k} \circ \cdots \circ B_{N_2}^{\varphi_2} \circ B_{N_1}^{\varphi_1}\right)(\mathbf{x}) \tag{B.1}$$

where $\varphi = (\varphi_1, \ldots, \varphi_{k+1})$. Note that the lift operation can be different in each of the k neural-lift-signature blocks $B_{N_k}^{\varphi_k}$.

Appendix C

Further experimental details for Part III

C.1 Posterior sampling

C.1.1 Sampling with Metropolis-Hastings

The Metropolis-Hastings algorithm is a classical algorithm for generating samples from some density $\pi(\boldsymbol{\theta})$. Starting from $\boldsymbol{\theta}^{(0)}$ and given a proposal distribution $q(\cdot \mid \boldsymbol{\theta})$ which proposes successive values in the chain, it entails repeating the following steps: at each step $t \geq 1$,

1. propose $\boldsymbol{\theta} \sim q(\cdot \mid \boldsymbol{\theta}^{(t)})$;
2. accept $\boldsymbol{\theta}$ (i.e. set $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}$) with probability

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta})}{\pi(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})}\right\},$$

   else set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$.

Since such a procedure generates correlated samples from the posterior, it is customary to *thin* the samples by retaining every nth sample for some integer $n \geq 1$ chosen as required.

To sample from the posteriors obtained via KDE and NRE in Section 6.2, we use Metropolis-Hastings (MH) with a normal proposal distribution $q(\cdot \mid \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \ell^2\Sigma)$, and perform a trial run of 50,000 steps using an isotropic Gaussian to estimate the covariance matrix Σ of q before a further 100,000 steps are run. The use of such pilot runs is long standing-practice for random walk MH algorithms and can be motivated theoretically, for example, using Bayesian asymptotics (Schmon and Gagnon, 2021). In addition, we set $\ell = 2/\sqrt{d}$, where d is the parameter dimension following the guidelines of Gelman et al. (1996); Roberts et al. (1997); Schmon and Gagnon (2021). All chains are initialised at the parameter value which generated the observation. We thin the resultant chains by retaining every 100th value, resulting in 1,000 approximately uncorrelated samples from the respective posteriors.

C.1.2 Sampling with sampling-importance-resampling

Sampling-importance-resampling (SIR) is an approach to obtaining samples from a target distribution $f(\boldsymbol{\theta})$ given samples from a different distribution $g(\boldsymbol{\theta})$ which proceeds as follows:

1. generate samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N} \overset{iid}{\sim} g(\boldsymbol{\theta})$;
2. compute weights $w_i = f(\boldsymbol{\theta}^{(i)})/g(\boldsymbol{\theta}^{(i)})$ for all i;
3. resample $\boldsymbol{\theta}^{(i)}$ with probability proportional to w_i.

In particular, this may be used in density ratio estimation to approximate the posterior by setting $g(\boldsymbol{\theta}) := \pi(\boldsymbol{\theta})$, $f(\boldsymbol{\theta}) := \pi(\boldsymbol{\theta} \mid \mathbf{y})$, and using that the density ratio estimator estimates the ratio

$$\frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})} = \frac{\pi(\boldsymbol{\theta} \mid \mathbf{y})}{\pi(\boldsymbol{\theta})} = \frac{p(\mathbf{y} \mid \boldsymbol{\theta})}{p(\mathbf{y})}. \tag{C.1}$$

C.2 Sampling with kernel density estimation & Markov chain Monte Carlo

To sample from the posteriors obtained via KDE, we use Metropolis-Hastings (MH) (see Appendix C.1.1 with a normal proposal distribution $q(\cdot \mid \cdot)$, and perform a trial

run of 50,000 steps to estimate the covariance matrix of q before a further 100,000 steps are run. All chains are initialised at the parameter value which generated the observation. We thin the resultant chains by retaining every 100th value, resulting in 1,000 approximately uncorrelated samples from the KDE posterior.

C.3 Graph Neural Networks

Recent years have seen considerable progress in the development of graph neural networks (GNNs) in machine learning (e.g. Yao et al., 2019; Zhang et al., 2020; Baek et al., 2021). In many cases, the design of a GNN consists of generalising a convolution operator from regular, Euclidean domains – as appears in convolutional neural networks – to graphs. This has predominantly proceeded by constructing a convolution in the spatial domain (see e.g. Masci et al., 2015; Niepert et al., 2016) or by exploiting the convolution theorem and performing a multiplication in the graph Fourier domain (see e.g. Bruna et al., 2014). A recent review of GNNs and their design can be found in Zhou et al. (2020).

The problem of extending GNNs to dynamic graphs has also recently received significant attention. In this vein, Li et al. (2017b) introduce Diffusion Convolutional Recurrent Neural Networks, with applications to traffic flow prediction. In addition, Seo et al. (2018) propose Graph Convolutional Recurrent Networks, an adaptation of standard recurrent networks to operate on sequences of graphs via graph convolutional operators. Further examples of recurrent graph neural network architectures exist; a broader survey of neural networks for dynamic graphs can be found in Wu et al. (2021, Section 7).

C.4 Neural network architectures and training

For each neural network method involving vector-valued time-series, the z-scores of all variables are taken prior to passing them into the networks.

For the graph embedding network in Section 6.3.2.2, the first module is a graph convolutional gated recurrent unit proposed in (Seo et al., 2018). Taking $L \in \mathbb{R}^{N \times N}$ as the normalised graph Laplacian for the order-N graph, this component operates on

the sequence $(\mathbf{x}_t)_{t=1}^T$ of node states $\mathbf{x}_t \in \mathbb{R}^{N\times K}$ – where $K \geq 1$ is the dimensionality of each node state – to find a running embedding $\mathbf{h}_t \in \mathbb{R}^{N\times d_\mathbf{h}}$ of each subsequence $(\mathbf{x}_{t'})_{t'=1}^t$, $t = 1, \ldots, T$, as follows:

$$\begin{aligned}
\mathbf{s} &= \sigma\left(W_{\mathbf{sx}}(L)\cdot\mathbf{x}_t + W_{\mathbf{sh}}(L)\cdot\mathbf{h}_{t-1}\right),\\
\mathbf{r} &= \sigma\left(W_{\mathbf{rx}}(L)\cdot\mathbf{x}_t + W_{\mathbf{rh}}(L)\cdot\mathbf{h}_{t-1}\right),\\
\tilde{\mathbf{h}} &= \tanh\left(W_{\mathbf{hx}}(L)\cdot\mathbf{x}_t + W_{\mathbf{hh}}(L)\cdot(\mathbf{r}\odot\mathbf{h}_{t-1})\right),\\
\mathbf{h}_t &= \mathbf{s}\odot\mathbf{h}_{t-1} + (1-\mathbf{s})\odot\tilde{\mathbf{h}}
\end{aligned}$$

for $t = 1, \ldots, T$. Here, $W_{\cdot\mathbf{x}} : \mathbb{R}^{N\times N} \to \mathbb{R}^{Q\times d_\mathbf{h}\times K}$ and $W_{\cdot\mathbf{h}} : \mathbb{R}^{N\times N} \to \mathbb{R}^{Q\times d_\mathbf{h}\times d_\mathbf{h}}$ are graph convolution operators with $d_\mathbf{h}K$ filters that are parameterised by Q Chebyshev coefficients, taking the form

$$G(L) = \sum_{q=1}^{Q} a_q H_q(L), \tag{C.2}$$

where $H_q(L)$ is the q-th order Chebyshev polynomial evaluated at L. We use $Q = 3$ Chebyshev coefficients in the graph filtering operation and choose a hidden state size of $d_\mathbf{h} = 64$, such that the hidden state of each agent is a 64-dimensional vector. A single linear layer reduces this $N \times 64$ matrix into an N-vector, where N is the number of agents in the system. An embedding of the entire graph then proceeds by passing this N-vector through a feedforward network with layer sizes $32, 16, 16$. In our experiments, we take $N = 50$ and simulate for $T = 25$ time steps.

For all neural posterior estimation tasks, we use a masked autoregressive flow (Papamakarios et al., 2017) with 5 flow transforms, each with 2 blocks and 50 hidden features; for all neural density ratio estimation tasks, we use a residual network with two layers of size 50. To train the network weights, we use Adam (Kingma and Ba, 2014), along with a training batch size of 50 and learning rate of 5×10^{-4}. We furthermore reserve 10% of the data for validation, and stop training when the validation error does not improve over 20 epochs to avoid overfitting. Throughout, we use the `sbi` python package (Tejero-Cantero et al., 2020) and PyTorch Geometric Temporal (Rozemberczki et al., 2021) `python` packages.

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

J. Aitchison. Goodness of Prediction Fit. *Biometrika*, 62(3):547–554, 1975. ISSN 00063444. URL http://www.jstor.org/stable/2335509.

Mattias Åkesson, Prashant Singh, Fredrik Wrede, and Andreas Hellander. Convolutional neural networks as summary statistics for approximate Bayesian computation. *arXiv preprint arXiv:2001.11760*, 2020.

Justin Alsing, Benjamin Wandelt, and Stephen Feeney. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. *Monthly Notices of the Royal Astronomical Society*, 477(3):2874–2885, Mar 2018. ISSN 1365-2966. doi: 10.1093/mnras/sty819. URL http://dx.doi.org/10.1093/mnras/sty819.

Justin Alsing, Tom Charnock, Stephen Feeney, and Benjamin Wandelt. Fast likelihood-free cosmology with neural density estimators and active learning. *Monthly Notices of the Royal Astronomical Society*, 488(3):4440–4458, 07 2019. ISSN 0035-8711. doi: 10.1093/mnras/stz1960. URL https://doi.org/10.1093/mnras/stz1960.

Christophe Andrieu, Gareth O Roberts, et al. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Jinheon Baek, Minki Kang, and Sung Ju Hwang. Accurate Learning of Graph Representations with Graph Multiset Pooling. In *ICLR*, 2021. URL https://openreview.net/forum?id=JHcqXGaqiGn.

Ruth E Baker, EA Gaffney, and PK Maini. Partial differential equations for self-organization in cellular and developmental biology. *Nonlinearity*, 21(11):R251, 2008.

Rafa Baptista, J Doyne Farmer, Marc Hinterschweiger, Katie Low, Daniel Tang, and Arzu Uluc. Macroprudential policy in an agent-based model of the UK housing market. 2016.

M. J. Bayarri and J. O. Berger. The Interplay of Bayesian and Frequentist Analysis. *Statistical Science*, 19(1):58 – 80, 2004. doi: 10.1214/088342304000000116. URL https://doi.org/10.1214/088342304000000116.

Mark A Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003.

Mark A Beaumont. Approximate Bayesian computation. *Annual review of statistics and its application*, 6:379–403, 2019.

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

Mark A. Beaumont, JEAN-MARIE CORNUET, JEAN-MICHEL MARIN, and CHRISTIAN P. ROBERT. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/27798882.

Espen Bernton, Pierre E. Jacob, Mathieu Gerber, and Christian P. Robert. Approximate Bayesian computation with the Wasserstein distance. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 81(2):235–269, 2019. ISSN 14679868. doi: 10.1111/rssb.12312.

Christopher M Bishop. Mixture density networks. 1994.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 78(5):1103, 2016.

D. Blackwell and R. V. Ramamoorthi. A Bayes but Not Classically Sufficient Statistic. *The Annals of Statistics*, 10(3):1025 – 1026, 1982. doi: 10.1214/aos/1176345895. URL https://doi.org/10.1214/aos/1176345895.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from Several Related Classification Tasks to a New Unlabeled Sample. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2178–2186, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

Michael GB Blum, Maria Antonieta Nunes, Dennis Prangle, Scott A Sisson, et al. A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science*, 28(2):189–208, 2013.

Horatio Boedihardjo, Xi Geng, Terry Lyons, and Danyu Yang. The signature of a rough path: Uniqueness. *Advances in Mathematics*, 293:720–737, 2016. ISSN 0001-8708. doi: https://doi.org/10.1016/j.aim.2016.02.011. URL https://www.sciencedirect.com/science/article/pii/S0001870816301104.

Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3): 7280–7287, 2002. ISSN 0027-8424. doi: 10.1073/pnas.082080899. URL https://www.pnas.org/content/99/suppl_3/7280.

Luke Bornn, Natesh S Pillai, Aaron Smith, and Dawn Woodard. The use of a single pseudo-sample in approximate Bayesian computation. *Statistics and Computing*, 27(3):583–590, 2017.

Elizabeth Bradley and Holger Kantz. Nonlinear time-series analysis revisited. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9):097610, Sep 2015. ISSN 1089-7682. doi: 10.1063/1.4917289. URL http://dx.doi.org/10.1063/1.4917289.

Johann Brehmer, Kyle Cranmer, Gilles Louppe, and Juan Pavez. Constraining Effective Field Theories with Machine Learning. *Phys. Rev. Lett.*, 121:111801, Sep 2018. doi: 10.1103/PhysRevLett.121.111801. URL https://link.aps.org/doi/10.1103/PhysRevLett.121.111801.

François Xavier Briol, Alessandro Barp, Andrew B. Duncan, and Mark Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv*, pages 1–57, 2019.

William A. Brock and Cars H. Hommes. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8):1235–1274, 1998. ISSN 0165-1889. doi: https://doi.org/10.1016/S0165-1889(98)00011-6. URL https://www.sciencedirect.com/science/article/pii/S0165188998000116.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and deep locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Patrick Cannon, Daniel Ward, and Sebastian M. Schmon. Investigating the Impact of Model Misspecification in Neural Simulation-based Inference, 2022. URL https://arxiv.org/abs/2209.01845.

Kuo-Tsai Chen. Integration of paths–a faithful representation of paths by noncommutative formal power series. *Transactions of the American Mathematical Society*, 89(2):395–407, 1958.

Yanzhi Chen, Dinghuai Zhang, Michael Gutmann, Aaron Courville, and Zhanxing Zhu. Neural Approximate Sufficient Statistics for Implicit Models. pages 1–14, 2020. URL http://arxiv.org/abs/2010.10079.

Ilya Chevyrev and Andrey Kormilitzin. A Primer on the Signature Method in Machine Learning. 2016. URL http://arxiv.org/abs/1603.03788.

Ilya Chevyrev and Harald Oberhauser. Signature moments to characterize laws of stochastic processes. *arXiv preprint arXiv:1810.10971*, 2018.

Ayush Chopra, Alexander Rodríguez, Jayakumar Subramanian, Balaji Krishnamurthy, B. Aditya Prakash, and Ramesh Raskar. Differentiable agent-based epidemiological modeling for end-to-end learning. In *ICML 2022 Workshop AI for*

Agent-Based Modelling, 2022. URL https://openreview.net/forum?id=9_7_zjAjSJF.

Kim Christensen, Kishan A. Manani, and Nicholas S. Peters. Simple model for identifying critical regions in atrial fibrillation. *Physical Review Letters*, 114(2):1–6, 2015. ISSN 10797114. doi: 10.1103/PhysRevLett.114.028104.

Kyle Cranmer, Juan Pavez, and Gilles Louppe. Approximating Likelihood Ratios with Calibrated Discriminative Classifiers, 2016.

Niccolo Dalmasso, Rafael Izbicki, and Ann Lee. Confidence Sets and Hypothesis Testing in a Likelihood-Free Inference Setting. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2323–2334. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/dalmasso20a.html.

Gokhan Danabasoglu, J-F Lamarque, J Bacmeister, DA Bailey, AK DuVivier, Jim Edwards, LK Emmons, John Fasullo, R Garcia, Andrew Gettelman, et al. The community earth system model version 2 (CESM2). *Journal of Advances in Modeling Earth Systems*, 12(2), 2020.

G Darmois. Sur les lois de probabilites a estimation exhaustive. In *C. R. Acad. Sci. Paris (in French)*, volume 200, page 1265–1266, 1935.

R Maria del Rio-Chanona, Penny Mealy, Mariano Beguerisse-Díaz, François Lafond, and J Doyne Farmer. Occupational mobility and automation: a data-driven network model. *Journal of the Royal Society Interface*, 18(174):20200898, 2021.

Peter J Diggle and Richard J Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):193–212, 1984.

Traiko Dinev and Michael U. Gutmann. Dynamic Likelihood-free Inference via Ratio Estimation (DIRE), 2018.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

Arnaud Doucet, Michael K Pitt, George Deligiannidis, and Robert Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015.

Christopher Drovandi and David T Frazier. A Comparison of Likelihood-Free Methods With and Without Summary Statistics. *arXiv preprint arXiv:2103.02407*, 2021.

Conor Durkan, Iain Murray, and George Papamakarios. On Contrastive Learning for Likelihood-free Inference. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2771–2781. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/durkan20a.html.

Joel Dyer and Blas Kolic. Public risk perception and emotion on Twitter during the Covid-19 pandemic. *Applied Network Science*, 5(1):1–32, 2020.

Joel Dyer, Patrick Cannon, and Sebastian M Schmon. Approximate Bayesian Computation with Path Signatures. *arXiv preprint arXiv:2106.12555*, 2021a.

Joel Dyer, Patrick W Cannon, and Sebastian M Schmon. Deep Signature Statistics for Likelihood-free Time-series Models. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021b.

Joel Dyer, Patrick Cannon, J Doyne Farmer, and Sebastian Schmon. Black-box Bayesian inference for economic agent-based models. *arXiv preprint arXiv:2202.00625*, 2022a.

Joel Dyer, Patrick Cannon, J Doyne Farmer, and Sebastian M Schmon. Calibrating Agent-based Models to Microdata with Graph Neural Networks. In *ICML 2022 Workshop AI for Agent-Based Modelling*, 2022b.

Joel Dyer, Patrick W. Cannon, and Sebastian M. Schmon. Amortised likelihood-free inference for expensive time-series simulators with signatured ratio estimation. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 11131–11144. PMLR, 28–30 Mar 2022c. URL https://proceedings.mlr.press/v151/dyer22a.html.

Joel Dyer, John Fitzgerald, Bastian Rieck, and Sebastian M Schmon. Approximate Bayesian Computation for Panel Data with Signature Maximum Mean Discrepancies. In *NeurIPS 2022 Temporal Graph Learning Workshop*, 2022d. URL https://openreview.net/forum?id=Bol45H5FAc.

Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6: 290–297, 1959.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

J Doyne Farmer and Duncan Foley. The economy needs agent-based modelling. *Nature*, 460(7256):685–686, 2009.

Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 74(3): 419–474, 2012. ISSN 13697412. doi: 10.1111/j.1467-9868.2011.01010.x.

R. A. Fisher and Edward John Russell. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1922. doi: 10.1098/rsta.1922.0009. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1922.0009.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Reiner Franke. Applying the method of simulated moments to estimate a small agent-based asset pricing model. *Journal of Empirical Finance*, 16(5):804–815, 2009.

Reiner Franke and Frank Westerhoff. Structural stochastic volatility in asset pricing dynamics: Estimation and model contest. *Journal of Economic Dynamics and Control*, 36(8):1193–1211, 2012.

Andrew M. Fraser and Harry L. Swinney. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, 33:1134–1140, Feb 1986. doi: 10.1103/PhysRevA.33.1134. URL https://link.aps.org/doi/10.1103/PhysRevA.33.1134.

David T Frazier, Christopher Drovandi, and Ruben Loaiza-Maya. Robust approximate Bayesian computation: An adjustment approach. *arXiv preprint arXiv:2008.04099*, 2020a.

David T Frazier, Christian P Robert, and Judith Rousseau. Model misspecification in approximate Bayesian computation: consequences and diagnostics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(2):421–444, 2020b.

David T Frazier, Christopher Drovandi, and David J Nott. Better together: pooling information in likelihood-free inference. *arXiv preprint arXiv:2212.02658*, 2022.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Andrew Gelman, Gareth O Roberts, Walter R Gilks, et al. Efficient Metropolis jumping rules. *Bayesian statistics*, 5(599-608):42, 1996.

Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977. doi: 10.1021/j100540a008. URL https://doi.org/10.1021/j100540a008.

Pedro J Gonçalves, Jan-Matthis Lueckmann, Michael Deistler, Marcel Nonnenmacher, Kaan Öcal, Giacomo Bassetto, Chaitanya Chintaluri, William F Podlaski, Sara A Haddad, Tim P Vogels, et al. Training deep neural density estimators to identify mechanistic models of neural dynamics. *Elife*, 9:e56261, 2020.

Christian Gourieroux, Alain Monfort, and Eric Renault. Indirect inference. *Journal of applied econometrics*, 8(S1):S85–S118, 1993.

Jakob Grazzini, Matteo G Richiardi, and Mike Tsionas. Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control*, 77:26–47, 2017.

David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic posterior transformation for likelihood-free inference. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4288–4304, 2019.

Rory Greig and Jordi Arranz. Generating Agent Based Models From Scratch With Genetic Programming. 2021.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19:513–520, 2006.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Lajos Gergely Gyurk. Extracting information from the signature of a financial data stream. pages 1–22, 2014.

Ben Hambly and Terry Lyons. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Annals of Mathematics*, 171(1):109–167, Mar 2010. ISSN 0003-486X. doi: 10.4007/annals.2010.171.109. URL http://dx.doi.org/10.4007/annals.2010.171.109.

Trevor Hastie, Robert Tibshirani, and J. H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, New York, 2001. ISBN 9780387952840.

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

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970b. ISSN 0006-3444. doi: 10.1093/biomet/57.1.97. URL https://doi.org/10.1093/biomet/57.1.97.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with Amortized Approximate Ratio Estimators. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4239–4248. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/hermans20a.html.

Nick Jagiella, Dennis Rickert, Fabian J Theis, and Jan Hasenauer. Parallelization and high-performance computing enables automated statistical inference of multi-scale models. *Cell systems*, 4(2):194–206, 2017.

Bai Jiang, Tung-yu Wu, Charles Zheng, and Wing H Wong. Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica*, pages 1595–1618, 2017.

Paul Joyce and Paul Marjoram. Approximately Sufficient Statistics and Bayesian Computation. *Statistical Applications in Genetics and Molecular Biology*, 7(1):1–18, August 2008. doi: 10.2202/1544-6115.1389. URL https://ideas.repec.org/a/bpj/sagmbi/v7y2008i1n26.html.

Nianqiao Ju, Jeremy Heng, and Pierre E Jacob. Sequential Monte Carlo algorithms for agent-based models of disease transmission. *arXiv preprint arXiv:2101.12156*, 2021.

L. V. Kantorovich. Mathematical Methods of Organizing and Planning Production. *Management Science*, 6(4):366–422, 1960. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2627082.

Cliff C Kerr, Robyn M Stuart, Dina Mistry, Romesh G Abeysuriya, Katherine Rosenfeld, Gregory R Hart, Rafael C Núñez, Jamie A Cohen, Prashanth Selvaraj, Brittany Hagedorn, et al. Covasim: an agent-based model of COVID-19 dynamics and interventions. *PLOS Computational Biology*, 17(7):e1009149, 2021.

Patrick Kidger, Patric Bonnier, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons. Deep signature transforms. *Advances in Neural Information Processing Systems*, 32, 2019.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *arXiv preprint arXiv:2005.08926*, 2020.

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

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Franz J. Kiŕaly and Harald Oberhauser. Kernels for sequentially ordered data. *Journal of Machine Learning Research*, 20:1–45, 2019. ISSN 15337928.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*, 2019.

A. N. Kolmogorov. Definition of center of dispersion and measure of accuracy from a finite number of observations (in russian). *Izv. Akad. Nauk S. S. S. R. Ser. Mat.*, 6:3–32, 1942. URL https://ci.nii.ac.jp/naid/10019644207/en/.

B. O. Koopman. On Distributions Admitting a Sufficient Statistic. *Transactions of the American Mathematical Society*, 39(3):399–409, 1936. ISSN 00029947. URL http://www.jstor.org/stable/1989758.

Adam Korobow, Chris Johnson, and Robert Axtell. An agent–based model of tax compliance with social networks. *National Tax Journal*, 60(3):589–610, 2007.

Jiri Kukacka and Jozef Barunik. Estimation of financial agent-based models with simulated maximum likelihood. *Journal of Economic Dynamics and Control*, 85: 21–45, 2017.

Finn E. Kydland and Edward C. Prescott. Time to Build and Aggregate Fluctuations. *Econometrica*, 50(6):1345–1370, 1982. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913386.

Theo Kypraios. Efficient Bayesian inference for partially observed stochastic epidemics and a new class of semi-parametric time series models. 2007.

Chenyang Li, Xin Zhang, and Lianwen Jin. LPSNet: A Novel Log Path Signature Feature Based Hand Gesture Recognition Framework. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018-January:631–639, 2017a. doi: 10.1109/ICCVW.2017.80.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017b.

Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, 2017.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking Simulation-Based Inference. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 343–351. PMLR, 13–15 Apr 2021.

Thomas Lux. Estimation of agent-based models using sequential Monte Carlo methods. *Journal of Economic Dynamics and Control*, 91:391–408, 2018.

Thomas Lux. Bayesian estimation of agent-based models via adaptive particle Markov Chain Monte Carlo. *Computational Economics*, pages 1–27, 2021.

Terry Lyons. Rough paths, signatures and the modelling of functions on streams. *arXiv preprint arXiv:1405.4537*, 2014.

Terry Lyons, Zhongmin Qian, et al. *System control and rough paths*. Oxford University Press, 2002.

Terry J Lyons, Michael Caruana, and Thierry Lévy. *Differential equations driven by rough paths : École d'été de probabilités de Saint-Flour XXXIV-2004 [electronic resource]*. Lecture notes in mathematics (Springer-Verlag); 1908. Springer, Berlin; New York, 2007. ISBN 9783540712855.

Michael W Macy, James A Kitts, Andreas Flache, and Steve Benard. Polarization in dynamic networks: A Hopfield model of emergent structure. *Dynamic Social Network Modelling and Analysis*, pages 162–173, 2003.

Nick Malleson, Kevin Minors, Le-Minh Kieu, Jonathan A Ward, Andrew West, and Alison Heppenstall. Simulating crowds in real time with agent-based modelling and a particle filter. *Journal of Artificial Societies and Social Simulation*, 23(3), 2020.

Jean Michel Marin, Pierre Pudlo, Christian P. Robert, and Robin J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012. ISSN 09603174. doi: 10.1007/s11222-011-9288-2.

Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Geodesic Convolutional Neural Networks on Riemannian Manifolds. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 832–840, 2015. doi: 10.1109/ICCVW.2015.112.

Jeffrey W Miller and David B Dunson. Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, 2018.

P. J. Moore, T. J. Lyons, and J. Gallacher. Using path signatures to predict a diagnosis of Alzheimer's disease. *PLoS ONE*, 14(9):1–16, 2019. ISSN 19326203. doi: 10.1371/journal.pone.0222212. URL http://dx.doi.org/10.1371/journal.pone.0222212.

James Morrill, Andrey Kormilitzin, Alejo Nevado-Holgado, Sumanth Swaminathan, Sam Howison, and Terry Lyons. The Signature-Based Model for Early Detection of Sepsis from Electronic Health Records in the Intensive Care Unit. *Computing in Cardiology*, 2019-Septe:2–5, 2019. ISSN 2325887X. doi: 10.23919/CinC49843.2019.9005805.

James Morrill, Adeline Fermanian, Patrick Kidger, and Terry Lyons. A Generalised Signature Method for Time Series. *arXiv preprint*, 2020.

Shigeki Nakagome, Kenji Fukumizu, and Shuhei Mano. Kernel approximate Bayesian computation in population genetic inferences. *Statistical Applications in Genetics and Molecular Biology*, 12(6):667–678, 2013. doi: doi:10.1515/sagmb-2012-0050. URL https://doi.org/10.1515/sagmb-2012-0050.

Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705 – 767, 2003. doi: 10.1214/aos/1056562461. URL https://doi.org/10.1214/aos/1056562461.

J. Neyman. Sur un teorema concernente le cosidette statistiche sufficienti. *Giorn 1st Ital Att*, 6:320–334, 1935.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. doi: 10.1126/science.aaw1147. URL https://www.science.org/doi/abs/10.1126/science.aaw1147.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with Bayesian conditional density estimation. In *Advances in neural information processing systems*, pages 1028–1036, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2335–2344, 2017.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR, 2019.

Mijung Park, Wittawat Jitkrittum, and Dino Sejdinovic. K2-ABC: Approximate Bayesian computation with kernel embeddings. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, 41:398–407, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Gabriel Peyre and Marco Cuturi. Computational Optimal Transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Kim Cuc Pham, David J Nott, and Sanjay Chaudhuri. A note on approximating abc-mcmc using flexible classifiers. *Stat*, 3(1):218–227, 2014.

Edwin James George Pitman. Sufficient statistics and intrinsic accuracy. In *Mathematical Proceedings of the cambridge Philosophical society*, volume 32, pages 567–579. Cambridge University Press, 1936.

Donovan Platt. A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, 113:103859, 2020. ISSN 0165-1889. doi: https://doi.org/10.1016/j.jedc.2020.103859. URL https://www.sciencedirect.com/science/article/pii/S0165188920300294.

Donovan Platt. Bayesian Estimation of Economic Simulation Models using Neural Networks. *Computational Economics*, pages 1–52, 2021.

Dennis Prangle. Summary statistics in approximate Bayesian computation. In Scott A Sisson, Yanan Fan, and Mark Beaumont, editors, *Handbook of approximate Bayesian computation*, pages 125–152. FL: CRC, 2018.

Leah F Price, Christopher C Drovandi, Anthony Lee, and David J Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

Gareth O Roberts, Andrew Gelman, and Walter R Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, , Guzman Lopez, Nicolas Collignon, and Rik Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, page 4564–4573, 2021.

Cristopher Salvi, Thomas Cass, James Foster, Terry Lyons, and Weixin Yang. The Signature Kernel is the solution of a Goursat PDE. *SIAM Journal on Mathematics of Data Science*, 3(3):873–899, 2021.

S M Schmon, G Deligiannidis, A Doucet, and M K Pitt. Large-sample asymptotics of the pseudo-marginal method. *Biometrika*, 108(1):37–51, 03 2021. ISSN 0006-3444. doi: 10.1093/biomet/asaa044. URL https://doi.org/10.1093/biomet/asaa044.

Sebastian M Schmon and Philippe Gagnon. Optimal scaling of random walk Metropolis algorithms using Bayesian large-sample asymptotics. *arXiv preprint arXiv:2104.06384*, 2021.

Sebastian M Schmon, Patrick W Cannon, and Jeremias Knoblauch. Generalized Posteriors in Approximate Bayesian Computation. *arXiv preprint arXiv:2011.08644*, 2020.

Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511809682.

Chris Sherlock, Alexandre H Thiery, Gareth O Roberts, and Jeffrey S Rosenthal. On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics*, 43(1):238–275, 2015.

Chris Sherlock, Alexandre H Thiery, and Anthony Lee. Pseudo-marginal Metropolis–Hastings sampling using averages of unbiased estimators. *Biometrika*, 104(3):727–734, 2017.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26. CRC Press, 1986.

Bruce Stephenson and Johannes Kepler. *Kepler's physical astronomy*. Princeton University Press, Princeton, 1994. ISBN 9780691036526.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Probabilistic Classification*, page 47–55. Cambridge University Press, 2012. doi: 10.1017/CBO9781139035613.007.

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

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

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian Inference Algorithms with Simulation-Based Calibration, 2020.

Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997.

Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020. doi: 10.21105/joss.02505. URL https://doi.org/10.21105/joss.02505.

Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U. Gutmann. Likelihood-Free Inference by Ratio Estimation. *Bayesian Analysis*, pages 1 – 31, 2021. doi: 10.1214/20-BA1238. URL https://doi.org/10.1214/20-BA1238.

Matthew Thorpe, Serim Park, Soheil Kolouri, Gustavo K Rohde, and Dejan Slepčev. A Transportation L^p Distance for Signal Analysis. *Journal of mathematical imaging and vision*, 59(2):187–210, 2017.

A. M. Turing. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952.

G.E. Uhlenbeck and L.S. Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823–841, 1930. ISSN 0031899X.

Tuong Manh Vu, Charlotte Probst, Joshua M Epstein, Alan Brennan, Mark Strong, and Robin C Purshouse. Toward inverse generative social science using multi-objective genetic programming. In *Proceedings of the genetic and evolutionary computation conference*, pages 1356–1363, 2019.

Tuong Manh Vu, Charlotte Buckley, Joao A Duro, and Robin C Purshouse. Exploring social theory integration in agent-based modelling using multi-objective grammatical evolution. In *ICML 2022 Workshop AI for Agent-Based Modelling*, 2022.

Daniel Wegmann, Christoph Leuenberger, and Laurent Excoffier. Efficient Approximate Bayesian Computation Coupled With Markov Chain Monte Carlo Without Likelihood. *Genetics*, 182(4):1207–1218, 08 2009. ISSN 1943-2631. doi: 10.1534/genetics.109.102509. URL https://doi.org/10.1534/genetics.109.102509.

Richard David Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology*, 12(2):129–141, 2013.

Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Proceedings of the 14th annual conference on Neural Information Processing Systems*, pages 682–688, 2001.

Samuel Wiqvist, Pierre-Alexandre Mattei, Umberto Picchini, and Jes Frellsen. Partially exchangeable networks and architectures for learning summary statistics in approximate Bayesian computation. In *International Conference on Machine Learning*, pages 6798–6807. PMLR, 2019.

Wing Wong, Bai Jiang, Tung-yu Wu, and Charles Zheng. Learning Summary Statistic for Approximate Bayesian Computation via Deep Neural Network. *Statistica Sinica*, 2018. ISSN 1017-0405. doi: 10.5705/ss.202015.0340. URL http://dx.doi.org/10.5705/ss.202015.0340.

Simon N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010. ISSN 00280836. doi: 10.1038/nature09319.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

Zecheng Xie, Zenghui Sun, Lianwen Jin, Hao Ni, and Terry Lyons. Learning Spatial-Semantic Context with Fully Convolutional Recurrent Network for Online Handwritten Chinese Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1903–1917, 2018. ISSN 01628828. doi: 10.1109/TPAMI.2017.2732978.

Tianbao Yang, Yu-feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström Method vs Random Fourier Features: A Theoretical and Empirical Comparison. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/621bf66ddb7c962aa0d22ac97d69b793-Paper.pdf.

Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377, 2019.

Xiao Zhang, Cristopher Moore, and Mark EJ Newman. Random graph models for dynamic networks. *The European Physical Journal B*, 90(10):1–14, 2017.

Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. Efficient Probabilistic Logic Reasoning with Graph Neural Networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJg76kStwH.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000012.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, December 1997. ISSN 0098-3500. doi: 10.1145/279232.279236. URL https://doi.org/10.1145/279232.279236.