University of Oxford

Mathematical Institute

Bayesian Inference for Economic Agent-based Models

Author:
Donovan Platt,
St. John's College

Supervisor:
Prof. Doyne Farmer



A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

September 2021

Acknowledgements

The completion of this thesis was made possible through the support, financial or otherwise, of a number of individuals, groups and organisations. I would therefore like to thank:

- My supervisor, Prof. Doyne Farmer, for professional and personal support provided throughout the various stages of this investigation, particularly during its most trying times.
- The various past and current members of the INET Oxford Complexity Economics research group, who allowed for the creation of a stimulating and supportive research environment, with particular thanks to Adrián Carro, Marco Pangallo, Blas Kolic, and Julian Winkler.
- My friends and family, for consistent emotional support and encouragement throughout the preceding years, with particular thanks to Yue Zhang, Matthew Mason, Hugh Bendor-Samuel, and the St. Aldates postgraduate community.
- My parents, for all that they have done for me throughout my life.
- The UK government, for the provision of a Commonwealth Scholarship, without which I would have been unable to undertake this research.
- Amazon Web Services, for the generous provision of access to cloud computing resources, which were ultimately used to generate a number of key results.
- BMLL Technologies, for the generous provision of access to limit order book data, which formed an important component of the analysis presented in Chapter 4.

In loving memory of Marie Rafferty
1942-2021

Publications

This thesis has yielded the following publications:

- Platt, D., 2020. A comparison of economic agent-based model calibration methods. J. Econ. Dyn. Control 113, 103859.
- Platt, D., 2021. Bayesian estimation of economic simulation models using neural networks. Comput. Econ. 10.1007/s10614-021-10095-9.

Additionally, Chapters 4 and 5 represent work that is expected to yield two additional publications at a later stage.

Abstract

Interest in agent-based models of financial markets and the wider economy has increased consistently over the last few decades, in no small part due to their ability to reproduce a number of empirically-observed stylised facts that are not easily recovered by more traditional modelling approaches. Nevertheless, the economic agent-based modelling paradigm faces mounting criticism, focused particularly on the rigour of current validation, calibration, and model selection practices, which still, in many cases, remain qualitative and stylised fact-driven. While the literature on quantitative and data-driven approaches has seen significant expansion in recent years, many studies have focused on the introduction of new methods that are neither benchmarked against existing alternatives nor rigorously tested on the large-scale and computationally-expensive variants that now dominate the literature, instead choosing to focus on highly-simplified models that are no longer representative of the current state of the art. This has ultimately led to some doubt regarding the applicability of current approaches to the kinds of problems of interest to most contemporary modellers and policymakers.

In response, this thesis attempts to make a number of meaningful contributions, both by providing a more complete and unified perspective on the existing literature and by additionally advancing it in a number of key areas. To elaborate, in Chapters 1 and 2, we review existing contributions and compare a number of prominent agent-based model estimation methods, both established and new, through the use of an experimental protocol that allows for the determination of the respective strengths and weaknesses of each approach and the quality of the resultant parameter estimates. Overall, we find that a simple, likelihood-based approach to Bayesian estimation consistently outperforms several members of the far more popular class of simulated minimum distance methods and results in reasonable parameter estimates in many contexts, with a degradation in performance observed only when considering a large-scale model and attempting to calibrate a substantial number of its parameters. This ultimately leads us to argue for a paradigm shift, suggesting that an increased emphasis on Bayesian methods is much needed and that attempts should be made to improve the performance of existing techniques.

Following on from the above recommendations, Chapter 3 proposes a neural network-based approach to Bayesian estimation that addresses a number of the fundamental weaknesses of the best performing methodology identified in Chapter 2. When benchmarking our proposed approach, we find that it consistently results in more accurate parameter estimates in a variety of settings, including the estimation of financial heterogeneous agent models, the identification of changes in dynamics occurring in models incorporating structural breaks, and applications to a large-scale, state-of-the-art model of the UK housing market. Chapter 4 further builds upon these ideas and demonstrates how the proposed Bayesian estimation methodology may be extended to facilitate the application of contemporary Bayesian model selection tools to a variety of inter- and intraday financial agent-based models and to benchmark such models against traditional asset return modelling approaches.

Finally, Chapter 5, which concludes the thesis, demonstrates how recent advances in Gaussian process emulation may be applied to the approaches developed in Chapters 3 and 4 to significantly reduce the overall computational costs and thus yield a computationally-efficient and widely-applicable framework for Bayesian inference in simulation models.

Contents

1 Literature Review **1**
- 1.1 Recent Progress in ABM Estimation and Calibration . . . 3
  - 1.1.1 Direct Observation . . . 3
  - 1.1.2 Analytical Methods . . . 4
  - 1.1.3 Simulation-based Methods . . . 4
  - 1.1.4 Shortcomings of the Literature . . . 9
- 1.2 Agent-based Approaches to Financial Market Modelling . . . 10
  - 1.2.1 A Brief Overview of Financial Markets . . . 11
  - 1.2.2 Interday Financial ABMs . . . 13
  - 1.2.3 Intraday Financial ABMs . . . 14

2 A Comparison of Economic Agent-based Model Calibration Methods **16**
- 2.1 Experimental Procedure . . . 16
  - 2.1.1 Loss Function Construction and Comparison Procedure . . . 17
  - 2.1.2 Implemented Models . . . 18
  - 2.1.3 Implemented Methods . . . 21
- 2.2 Results and Discussion . . . 26
  - 2.2.1 Simple Time Series Models . . . 26
  - 2.2.2 INET Oxford Housing Market Model . . . 42
- 2.A Additional Experimental Details . . . 47
  - 2.A.1 Simple Time Series Models . . . 48
  - 2.A.2 INET Oxford Housing Market Model . . . 50
- 2.B Sampling Algorithm Modifications . . . 53
- 2.C Computational Time Experiments . . . 55

3 Bayesian Estimation of Economic Simulation Models using Neural Networks **58**
- 3.1 Estimation and Experimental Procedures . . . 58
  - 3.1.1 Bayesian Estimation of Simulation Models . . . 59
  - 3.1.2 The Approach of Grazzini et al. (2017) . . . 59
  - 3.1.3 Likelihood Approximation using Neural Networks . . . 60
  - 3.1.4 Method Comparison and Benchmarking . . . 61
- 3.2 Candidate Models . . . 62
  - 3.2.1 Revisited Models . . . 63
  - 3.2.2 Franke and Westerhoff (2012) Model . . . 63
  - 3.2.3 AR(2)-GARCH(1, 1) Model . . . 64

3.3 Results and Discussion . . . 64
3.3.1 Comparative Exercises . . . 64
3.3.2 Empirical Examples . . . 74
3.4 Practical Considerations . . . 79
3.4.1 Choosing the Lag Length . . . 79
3.4.2 Computational Costs and Applicability to Large-scale Models . . . 82
3.A Technical Details of the Proposed Estimation Procedure . . . 86
3.A.1 Training Set Construction . . . 86
3.A.2 Neural Network Specification and Training . . . 86
3.A.3 Data Normalisation and Regularisation . . . 87
3.A.4 Neural Network Architecture . . . 88
3.B Supplementary Brock and Hommes (1998) Model Exercises . . . 89
3.C Robustness Tests . . . 90
3.C.1 Number of Lags . . . 91
3.C.2 Neural Network Architecture . . . 91
3.C.3 Neural Network Random Seed . . . 94

4 Bayesian Model Selection for Inter- and Intraday Financial Agent-based Models 96
4.1 Bayesian Predictive Model Selection . . . 97
4.1.1 The Deviance Information Criterion . . . 97
4.1.2 The Watanabe-Akaike Information Criterion . . . 99
4.1.3 Pareto Smoothed Importance Sampling Leave-one-out Cross-validation 100
4.2 Experimental Procedure . . . 104
4.2.1 Likelihood Approximation and Bayesian Estimation . . . 104
4.2.2 Simulation-based Validation . . . 105
4.3 Candidate Models . . . 107
4.3.1 Interday Models . . . 107
4.3.2 Intraday Models . . . 109
4.4 Results and Discussion . . . 115
4.4.1 Interday Models . . . 115
4.4.2 Intraday Models . . . 119
4.A Parameter Priors and Pseudo-empirical Data Parameterisation . . . 125
4.B PSIS-LOO Reliability Diagnostics . . . 127
4.C Order Book Data Preprocessing . . . 127
4.C.1 BMLL Data Lab . . . 128
4.C.2 Downsampling . . . 128
4.C.3 Illustrations . . . 130
4.D Bayesian Forecasting . . . 130
4.D.1 An Illustrative Example: The S&P 500 . . . 132

5 Gaussian Process-emulated Adaptive Markov Chain Monte Carlo for Economic Simulation Models 135
5.1 Surrogate Modelling . . . 135
5.1.1 Weaknesses of Global Surrogate Modelling . . . 136
5.1.2 Gaussian Processes . . . 137
5.2 The Approach of Järvenpää and Corander (2021) . . . 139
5.2.1 The Surrogate Log-likelihood . . . 140

5.2.2 The Metropolis-Hastings Acceptance Decision . . . 141
5.2.3 Retraining the Surrogate . . . 142
5.2.4 Full Algorithm and Modifications . . . 142
5.3 Results and Discussion . . . 144
5.3.1 Random Walk Model . . . 144
5.3.2 Brock and Hommes (1998) Model . . . 146
5.3.3 INET Oxford Housing Market Model . . . 148
5.4 Practical Considerations . . . 150
5.A Total Variation Distance . . . 154
5.B Additional Experimental Details . . . 154
5.B.1 Dealing with Invalid Likelihood Evaluations . . . 154
5.B.2 Rescaling of the Training Data . . . 155
5.B.3 Optimisation . . . 155

Chapter 1

Literature Review

The modelling of economic systems presents a significant challenge – the heterogeneity of their constituent agents, the complex nature of the interactions within them, and the non-linearity of their emergent dynamics, makes them extremely difficult, if not impossible to represent using analytical methods. As a result, the use of strong assumptions[1] to facilitate mathematical tractability has for many decades been the dominant paradigm in economic modelling (Geanakoplos and Farmer 2008; Farmer and Foley 2009; Fagiolo and Roventini 2017).

Unfortunately, many of the aforementioned assumptions, while mathematically-convenient, are also largely empirically-inconsistent. Take, for example, the representative agent assumption, a hallmark of dynamic stochastic general equilibrium (DSGE) models, which are ubiquitous in the literature (Bianchi et al. 2007; Fagiolo and Roventini 2017). Under this assumption, the behaviours of individual agents can be aggregated into a single, representative agent – a classic example of reductionist thinking (Bianchi et al. 2007). There exists a plethora of evidence, however, that this is likely not the case (Sonnenschein 1973; Mantel 1974; Debreu 1974). Rather, it turns out that the properties of multiple, interacting agents cannot be sufficiently approximated by simple averaging and that heterogeneity is indeed a key feature of real economies. It therefore follows that many traditional models, and particularly DSGE models, are likely built on dubious theoretical grounds[2].

Such criticisms of traditional economic modelling paradigms have become increasingly prominent in the wake of the Great Recession of the late 2000s, where the heterogeneity of agents in housing markets is believed to have played an important role and during which DSGE models were largely unable to provide meaningful insights (Farmer and Foley 2009; Geanakoplos et al. 2012; Fagiolo and Roventini 2017). In this context, the need for a viable alternative is clear.

Recent advances in computing power, along with successes in other domains such as ecology, have resulted in the emergence of a growing community arguing that agent-based models (ABMs), which simulate systems at the level of individual agents and the interactions between them[3], may provide a more principled approach to the modelling of the economy (Farmer and Foley 2009; Fagiolo and Roventini 2017). Indeed, recent decades have seen the

[1] These include, but are not limited to: the perfect rationality of agents, the existence of representative agents, and the efficient market hypothesis.

[2] See the survey by Fagiolo and Roventini (2017) for a more detailed critique of DSGE models.

[3] See Macal and North (2010) for a brief introduction to agent-based modelling.

emergence of a wide variety of economic and financial ABMs that largely dispense with the unrealistic assumptions that characterise traditional approaches in favour of more realistic alternatives rooted in empirically-observed behaviours (Chen 2003; LeBaron 2006).

This paradigm shift has ultimately resulted in a degree of success. In more detail, ABMs are well-known for their ability to replicate empirically-observed stylised facts (qualitative properties that appear consistently in empirically-measured data), despite the fact that such properties are not readily recovered using traditional approaches (LeBaron 2006; Barde 2016). As an example, empirically-observed asset returns are typically characterised by a fat-tailed distribution, demonstrate no serial autocorrelation, and present evidence of volatility clustering (Cont 2001), phenomena which are now recovered by a wide variety of financial ABMs of varying degrees of sophistication. Similarly, in a broader economic context, recent years have seen the emergence of ABMs that are increasingly ambitious in scope[4] and capable of reproducing increasingly large sets of micro- and macroeconomic stylised facts, such as the Zipf distribution of firm sizes (Axtell 2001) and fat-tailed firm growth rates (Dosi et al. 2017), among many other similar properties.

Despite the aforementioned successes, however, ABMs face strong criticisms of their own, focused particularly on the inadequacy of current empirical validation, calibration, and model selection practices (Grazzini and Richiardi 2015). Even today, many studies introducing new models, particularly large-scale, macroeconomic simulators, employ validation procedures that are qualitative in nature and seldom venture beyond the demonstration of a candidate model's ability to reproduce a set of empirically-observed stylised facts (Panayi et al. 2013; Guerini and Moneta 2017). Calibration in such investigations is equally rudimentary, and typically takes the form of manual parameter adjustments or ad-hoc processes that aim to select parameters that allow a candidate model to reproduce the set of stylised facts considered during validation[5]. Model selection, if performed at all, might consist of little more than demonstrating that a particular model reproduces the largest number of known stylised facts.

While such stylised fact-centric methodologies may have served a reasonable purpose in the past, particularly given the focus of early ABM literature on replicating stylised facts, they have also ultimately limited the utility of economic ABMs. Without access to tools that allow for rigorous statistical inference and the routine fitting of ABMs to empirical data, ABMs remain, at best, qualitative tools. Furthermore, the very large number of models able to recover virtually identical numbers of stylised facts, the wide variety of behavioural rules employed in these models, and their resultant flexibility, have led to an increasingly broad literature of competing model specifications that have become increasingly difficult to discriminate between[6] (LeBaron 2006; Barde 2016; Lamperti et al. 2018).

In response to these criticisms, a small, but growing literature dealing with more sophisticated quantitative validation and calibration techniques has emerged (Fagiolo et al. 2019). While significant progress has been made, particularly in the last three years, this literature has not been without its own problems and thus forms a fertile area in which to make a meaningful contribution.

[4]See, for example, the Eurace (Cincotti et al. 2010) and Schumpeter Meeting Keynes (Dosi et al. 2010) models.

[5]The "indirect calibration" procedure employed by Jacob Leal et al. (2016) is a representative example.

[6]This is often referred to as the "wilderness of bounded rationality" problem (Geanakoplos and Farmer 2008; Barde 2016).

1.1 Recent Progress in ABM Estimation and Calibration

As previously alluded to, there is a significant overlap between the ABM validation and calibration literatures. In most cases, calibration is concerned with selecting model parameters that result in dynamics that are as close as possible to those observed in a particular dataset, measured according to some criterion[7]. The same criterion may then also be used for validation purposes. Therefore, while we focus exclusively on the problem of ABM calibration in this section, many of these discussions would also be applicable to ABM validation.

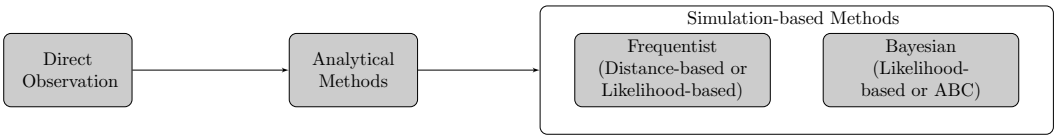


Figure 1.1: An illustration of the various calibration strategies one might consider when attempting to calibrate a particular ABM.

At this point, it is also worth noting that some authors use the terms calibration and estimation interchangeably, while others maintain that there are nuanced differences between them. As an example, Grazzini and Richiardi (2015) suggest that calibration is concerned only with obtaining agreement between the dynamics of a model and those of real-world data, while estimation additionally aims to ensure that the obtained model parameter values are an accurate reflection of the parameter values associated with the real-world data-generating process. Estimation would thus place additional emphasis on the uncertainty surrounding the obtained parameter values, while this may not be of much concern in the case of calibration. As stated by Hansen and Heckman (1996), however, the distinction between these terms is often inconsistent and not entirely clear. We will therefore use the terms interchangeably throughout this thesis, since the methods employed are likely to be very similar in either case.

Having now dealt with possible ambiguities in terminology, we proceed with a comprehensive review of the ABM calibration literature[8], beginning with the categorisation of ABM calibration strategies into three distinct classes, illustrated in Figure 1.1, which we then discuss in more detail.

1.1.1 Direct Observation

Since ABMs model systems by directly simulating the interactions of their microconstituents, it often arises that a number of model parameters reflect directly-observable (or easily-inferable) quantities, such as the net worth of firms or the distribution of ages among homeowners in a particular country. In this case, sophisticated estimation techniques are not required and the parameter values can be easily read directly from the data.

For some relatively simple models, such as the CATS model considered by Bianchi et al. (2007), values for almost all of the model's parameters can be determined in this way.

[7] These could include a set of stylised facts or some objective function.

[8] The interested reader may also wish to refer to the surveys of Lux and Zwinkels (2017) and Fagiolo et al. (2019).

While the same is not true for more sophisticated models, it still often arises that a large number of parameter values can be determined using a similar strategy, such as in the UK housing market model proposed by Baptista et al. (2016), where this process is referred to as micro-calibration.

1.1.2 Analytical Methods

Most ABMs include behavioural rules that require parameters that do not represent directly-observable quantities. Therefore, even if appropriate values for many of a model's parameters can be determined through direct observation, it would still be necessary to apply statistical estimation techniques to a given dataset to select appropriate values for the remaining parameters. A logical first choice might be maximum likelihood estimation, or alternatively the method of moments, given that they are fairly general and well-understood methods.

In a very limited number of cases, maximum likelihood estimation has been applied successfully to ABMs (Alfarano et al. 2005; 2006; 2007), but in general it and related methods are not appropriate as they rely on obtaining an analytical expression of the joint density function of time series simulated by the model. This is possible only for very simple models and even then is nontrivial.

1.1.3 Simulation-based Methods

Owing to the fact that most ABMs of interest do not have tractable likelihood functions and are incompatible with methods requiring analytical solutions for key model properties, it is inevitable that methods involving the generation of simulated data will need to be considered. Such approaches represent the key focus of most recent literature and we thus discuss them and related issues in extensive detail.

1.1.3.1 Traditional Approaches to Simulated Minimum Distance

The vast majority of existing calibration attempts have adopted a frequentist approach, often employing variations of what are commonly referred to as simulated minimum distance (SMD) methods (Grazzini and Richiardi 2015; Grazzini et al. 2017). Broadly speaking, these methods involve the construction of an objective function that measures the distance between simulated and measured time series for a given set of parameters, followed by the application of optimisation methods. More precisely, we have

$$\text{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} f(\boldsymbol{\theta}), \tag{1.1}$$

where $\boldsymbol{\theta}$ is a vector of parameters, $\boldsymbol{\Theta}$ is the space of feasible parameters, and $f : \boldsymbol{\Theta} \to \mathbb{R}$ is a function measuring the distance between real and simulated time series.

While truly general and standardised methods are yet to appear in the context of economic ABMs, methods which consider weighted sums of the squared errors between simulated and empirically-measured moments (or summary statistics) rose to prominence in early literature (Gilli and Winker 2003) and the well-known method of simulated moments[9] (MSM) has been applied in numerous investigations[10]. Key motivations for the consider-

[9] See Section 2.1.3.1 for a more detailed discussion on MSM.

[10] See Franke (2009), Franke and Westerhoff (2012), Fabretti (2013), Grazzini and Richiardi (2015), Chen and Lux (2018) and Platt and Gebbie (2018) for examples.

ation of MSM include its prevalence throughout econometric literature, its transparency (Franke 2009), and its well-understood mathematical properties[11].

MSM is not exempt from criticism, however. Its greatest shortcoming stems from the fact that the selection of moments is entirely arbitrary and a given set of moments will only represent a limited number of aggregate properties of the data and may therefore not sufficiently capture important dynamics. For this reason, estimation performance is strongly determined by the sufficiency of the chosen summary statistics and it has become increasingly apparent in recent literature that choosing an appropriate set is a difficult undertaking for financial and economic data (Grazzini et al. 2017; Chen and Lux 2018; Lux 2018).

A related approach is indirect inference (II), introduced by Gourieroux et al. (1993)[12]. Rather than relying on estimated moments, II involves the use of what is called an auxiliary model, essentially a simple model that is amenable to estimation via analytical methods such as maximum likelihood. An objective function is constructed by estimating the auxiliary model on both empirically-measured and simulated data and comparing the obtained parameters, with minimisation implying the greatest similarity between the two sets of estimated parameters. In general, II faces similar criticisms to those levelled against MSM (Grazzini et al. 2017), since it involves the selection of an arbitrary auxiliary model.

1.1.3.2 New Approaches to Simulated Minimum Distance

As previously stated, traditional approaches to SMD, such as MSM and II, suffer from a number of weaknesses, the most significant of which is the need to select an arbitrary set of moments or auxiliary model. Not surprisingly, a number of alternatives have emerged aiming to address this particular problem, focusing on the structure of a given time series and its patterns in an attempt to exploit the full informational content of the data and bypass any requirement to condense it into a set of summary statistics.

Among the most straightforward of these alternatives is choosing the objective function to be the sum of appropriately-weighted squared differences between the values of the simulated and empirically-measured time series at a number of points in time, as has been done by Recchioni et al. (2015). More sophisticated approaches to quantifying differences in the structure of simulated and empirically-measured time series are presented by Barde (2017) and Lamperti (2018a) in the form of information-theoretic distance measures called the Markov information criterion (MIC) and generalised subtracted L-divergence (GSL-div) respectively[13]. Recent years have also seen the emergence of attempts at comparing the causal mechanisms underlying real and simulated data through the use of SVAR regressions, as suggested by Guerini and Moneta (2017).

While the above approaches largely succeed in attenuating concerns related to the selection of arbitrary moments or auxiliary models and have received significant attention from the economic ABM community, they introduce new challenges. While MSM and II have well-understood theoretical properties, many of the recently introduced alternatives have not been subjected to rigorous mathematical analyses[14] (Grazzini et al. 2017). As a consequence, comparisons between different calibration methods tend to be non-trivial, as

[11] The estimator is both consistent and asymptotically normal for a well-chosen set of moments (McFadden 1989).

[12] An example of the application of II in the context of ABMs is presented by Bianchi et al. (2007).

[13] See Section 2.1.3.1 for a more detailed discussion on both the MIC and GSL-div.

[14] Some theoretical insights are available for the GSL-div and MIC, such as the expectation of the bias introduced by Jensen's inequality when estimating the required information-theoretic quantities, though a detailed analysis of the consistency and bias of the parameter estimates themselves is not currently available.

many of the aforementioned techniques have few conceptual similarities, and there have thus historically been few, if any, widely-accepted arguments in favour of a particular method.

1.1.3.3 Likelihood-based and Bayesian Approaches

Though the literature has historically been dominated by SMD methods, some newer investigations have instead attempted to produce simulation-based approximations to the likelihood function, such as the work of Kukacka and Barunik (2017), which employs a simple, non-parametric approach, and the work of Lux (2018), which investigates particle filters[15]. In both cases, the conducted experiments primarily employ the likelihood within a frequentist framework, similar to that of SMD methods[16].

Interestingly, Bayesian techniques have historically not generated much interest. This is perhaps rather surprising, given the plethora of Bayesian methods available for DSGE models (Fagiolo and Roventini 2017). The first major Bayesian study was conducted by Grazzini et al. (2017) and makes use of a relatively simple kernel density estimation-based likelihood approximation[17]. Though this investigation has become relatively well-cited and was well-received by the community, the Bayesian literature remained relatively sparse for quite some time after its initial release.

The past year has seen the emergence of a rather significant paradigm shift, however, following a number of interesting new contributions that have renewed interest in Bayesian methods. These include the comparative study of Platt (2020), which demonstrates the ability of the approach of Grazzini et al. (2017) to outperform several SMD alternatives[18], the probabilistic programming[19] study conducted by Bertschinger and Mozzhorin (2020), which makes use of the *Stan* language to perform a calibration and model selection exercise on similar models to those considered by Lux (2018), the work of Delli Gatti and Grazzini (2020), which applies similar techniques to those developed by Grazzini et al. (2017) to a medium-scale macroeconomic model[20], the work of Lux (2021), which applies the methodology considered by Lux (2018) to additional Bayesian examples involving several small-scale financial models, the work of Platt (2021), which introduces a neural network-based likelihood approximation method and benchmarks it against the approach of Grazzini et al.

[15] Much like newer approaches to SMD, likelihood-based approaches also dispense with the need to choose auxiliary models or summary statistics, making them a compelling option.

[16] Lux (2018) presents a single Bayesian estimation example, though the remainder of the investigation is largely frequentist.

[17] We provide a more thorough discussion on the approach of Grazzini et al. (2017) and Bayesian methods in general in Section 3.1.1.

[18] This work is presented in Chapter 2 of this thesis.

[19] Probabilistic programming is a relatively new paradigm in which a programmer specifies a model of interest in terms of a collection of random variables and the relationships between them using a specialised syntax. Once the model has been specified, Bayesian inference is carried out automatically by the language itself, which may be possible even if the model of interest lacks a tractable likelihood function, implying that such languages may be a possible option for ABMs. Such languages, have, however, historically only been applied to relatively simple models (in comparison to the large-scale economic ABMs that now dominate the literature) and the applicability of such approaches to more challenging inference problems remains up for debate, particularly given the difficulties associated with expressing more complex models in their restrictive syntax. We refer the interested reader to the work of van de Meent et al. (2018) for a detailed introduction to probabilistic programming.

[20] While Bayesian in nature, this work focuses on maximum a posteriori (MAP) estimation rather than the sampling of full posteriors. Additionally, several small improvements are also introduced, though these are not as comprehensive as those presented in Chapter 3 of this thesis, which was also originally made public at an earlier stage.

(2017) on both small- and large-scale ABMs[21], and finally the work of Shiono (2021), which attempts to estimate a medium-scale macroeconomic ABM via a procedure that bypasses the need for a likelihood and approximates the posterior directly using invertible neural networks.

1.1.3.4 Approximate Bayesian Computation

Up until this point, we have focused exclusively on the literature directly related to the problem of economic ABM estimation. Of course, this by no means implies that the associated difficulties, such as a lack of tractable likelihood functions, are exclusive to this domain. Indeed, it turns out that similar challenges are also frequently encountered in the field of computational biology. In response, a rich literature of likelihood-free methods for Bayesian inference, known as approximate Bayesian computation (ABC), has developed over the last few decades, beginning with the seminal contribution of Tavare et al. (1997).

In its canonical form, ABC replaces the likelihood with an alternative construction, $d(\boldsymbol{\mu}, \boldsymbol{\mu}^{sim}(\boldsymbol{\theta}))$, or more precisely, some form of distance[22] measuring the error between a set of summary statistics estimated from empirical data, $\boldsymbol{\mu}$, and an equivalent set estimated from data simulated by the candidate model, $\boldsymbol{\mu}^{sim}(\boldsymbol{\theta})$, for some proposed parameter set, $\boldsymbol{\theta}$. This measure can then be used, along with methods such as rejection sampling, to sample the posterior over the model parameters by randomly drawing samples from the prior and replacing the usual likelihood-based acceptance condition with $d(\boldsymbol{\mu}, \boldsymbol{\mu}^{sim}(\boldsymbol{\theta})) < \epsilon$.

As pointed out by Grazzini et al. (2017), ABC can essentially be viewed as a Bayesian version of MSM, where, instead of directly optimising $d(\boldsymbol{\mu}, \boldsymbol{\mu}^{sim}(\boldsymbol{\theta}))$, we embed it within a Bayesian framework and attempt to sample an approximate, summary statistic-based posterior[23]. Of course, this also implies that ABC methods inherit much of the disadvantages associated with MSM and its arbitrary selection of summary statistics. Additionally, the choice of the acceptance tolerance, ϵ, has a strong impact on estimation performance. If chosen to be too large, the resultant posterior approximation will be extremely coarse. Conversely, if chosen to be too small, the acceptance rate for newly proposed samples will be extremely low, resulting in a significant computational burden.

Despite the above disadvantages, however, ABC remains a dominant paradigm in computational biology and an active area of research, with some limitations, such as the need to select an acceptance threshold, being overcome in recent contributions[24]. The interested reader may therefore wish to refer to the authoritative text by Sisson et al. (2018) and the survey by Carrella (2021) for a more detailed summary of recent developments. Nevertheless, the need to condense data into a set of summary statistics remains a significant obstacle and a likely reason for the lack of any widespread adoption of ABC methods within the domain of economic agent-based modelling. Indeed, Grazzini et al. (2017) find that directly constructing a summary statistic-free likelihood approximation using kernel density estimation (KDE) delivers greater precision when compared to standard ABC methods, further limiting their appeal in our context[25]. For this reason, we do not directly consider ABC in

[21] This work is presented in Chapter 3 of this thesis.

[22] Unweighted or weighted forms of the Euclidean distance are most frequently used.

[23] This is in contrast to the methods discussed in Section 1.1.3.3, which attempt to directly approximate the likelihood as opposed to using a summary statistic-based alternative.

[24] See, for example, Sheehan and Song (2016) and Papamakarios and Murray (2016).

[25] We believe that this likely points to key domain differences. In economics and finance, data is often presented in the form of structured time series that may exhibit strong temporal trends, including structural breaks, crashes, and cyclical behaviour. Finding a set of summary statistics that can accurately capture these

this thesis and instead focus primarily on summary statistic-free methods.

1.1.3.5 Model Selection

In addition to the problem of parameter estimation, a key component of the statistical inference pipeline is the related task of model selection, which involves the determination of the most appropriate representation of the true data-generating process from a set of competing models[26]. This is particularly important in the context of ABMs, where robust model selection exercises incorporating a wide variety of traditional and agent-based modelling approaches could be used to make a case for the widespread adoption of ABMs.

In most cases, the task of discriminating between models inevitably leads to the construction of some measure of similarity or dissimilarity between model-simulated and empirically-measured data, which naturally implies strong parallels with many SMD methods. It is thus not surprising that many of the previously discussed approaches have also been applied in the model selection context, motivating the inclusion of this brief diversion in a section otherwise focused exclusively on calibration[27]. To elaborate, Franke and Westerhoff (2012) propose several variants of a novel financial heterogeneous agent model and make use of an MSM-based approach to select the best performing candidate. Using the MIC, Barde (2016) pits several prominent financial ABMs against each another and traditional time series models in a comprehensive series of head-to-head tests. Finally, Lamperti (2018b), who employs the GSL-div, compares several variants of the Brock and Hommes (1998) model.

While the above investigations do provide a number of important insights, it must be noted, however, that they do fall short in some areas. Firstly, regarding the techniques considered by Franke and Westerhoff (2012), standard criticisms of MSM and its arbitrary selection of moments apply. Secondly, it should be noted that, as a general practice, models should first be rigorously fit to the considered empirical data before being compared. Unfortunately, neither Barde (2016) nor Lamperti (2018b) carry out such a procedure, most likely due to computational constraints, and instead simply compare the candidate models at a fixed number of parameter combinations. Additionally, even if the models had been rigorously fit to the data using the MIC and GSL-div respectively as calibration objective functions, neither approach provides a proven method to penalise model complexity, which is especially relevant when models are fit and then scored against each other using the same set of empirical data[28].

To the best of our knowledge, the most developed attempt at model selection in the context of economic ABMs is presented in the previously discussed Bayesian investigation conducted by Bertschinger and Mozzhorin (2020), which considers similar models to Barde (2016) and deals with the issues of rigorous estimation and the penalisation of model complexity directly[29].

---

complex features is a non-trivial exercise. In contrast, it appears, at least from an outsider's perspective, that genetic and related datasets in computational biology are more amenable to being condensed into a set of summary statistics.

[26] Our discussion is kept intentionally brief; a far more detailed discussion of model selection and related issues is provided in Chapter 4.

[27] Interestingly, both the MIC and GSL-div were originally introduced as model selection methods, but generated widespread interest due to their potential to be employed as calibration methods.

[28] In other words, only in-sample performance is investigated; the same dataset is used to fit the models as is used to compare each model's relative performance.

[29] See Section 4.1.3.1 for a more detailed discussion.

1.1.4 Shortcomings of the Literature

From the preceding subsections, it should be apparent that the problem of ABM calibration has indeed become an area of greater focus in recent years. Nevertheless, we still believe that a number of fundamental shortcomings have limited the overall impact of the literature. We now discuss these issues in more detail.

1.1.4.1 Over-compartmentalisation

Having historically been a relatively niche area of research, past contributions introducing new approaches to ABM calibration have tended to be presented in a compartmentalised fashion. By this, we mean that new methods are typically presented without being strongly linked to prior literature and are also seldom benchmarked against existing alternatives. This, combined with the fact that the theoretical properties[30] of many newer techniques are not well understood (Grazzini et al. 2017), leaves the modeller with a difficult choice between a large number of methods with no obvious reason to favour one approach over another. As a result, many techniques have not been employed outside the context of the original manuscripts introducing them and have thus not been tested on or applied to new problems.

In this case, there are many lessons to be learnt from the field of machine learning, which has a strong culture of benchmarking and emphasises the creation of a core set of problems that new methodologies ought to be tested on. As an example, both the MNIST (LeCun et al. 1998) and CIFAR-10 (Krizhevsky 2009) datasets are standard test cases used to challenge image classifiers and the use of these common problem sets has allowed performance, speed, and other differences between new and existing methods to be more clearly understood.

We strongly believe that the above principles are essential for the advancement of the ABM calibration literature and that there is a significant need for benchmarking and the creation of standard test cases.

1.1.4.2 Computational Limitations

While the lack of comparisons between newer methods is indeed a significant weakness of the ABM calibration literature, computational difficulties remain an even larger obstacle in the development of robust and widely-applicable ABM calibration strategies. Since most models of interest are likely to be costly to simulate and given the large amount of data that needs to be generated for comparison in most ABM calibration frameworks, few attempts have been made to calibrate large-scale ABMs. Instead, most investigations favour proof-of-concept demonstrations on far simpler models that are essentially costless to simulate, such as the those of Brock and Hommes (1998) and Farmer and Joshi (2002). Ultimately, and despite conceptual similarities between large-scale models and simpler variants, the extent to which existing results can be generalised to state-of-the-art, large-scale models remains an open question[31]. There is thus an urgent need to test and refine existing and new approaches

[30] In most cases, this would refer to the bias and consistency of the associated estimators, which are difficult (or some perhaps even impossible to determine) due to the simulation-based nature of most methods.

[31] In fact, a number of new methodologies, such as the sequential Monte Carlo and probabilistic programming approaches of Lux (2018) and Bertschinger and Mozzhorin (2020) respectively, are incompatible with all but the simplest of model specifications. This is due to rapidly increasing computational costs relative to the number of latent variables in the case of the approach of Lux (2018) and a fundamental incompatibility with models containing discrete agents in the case of the approach of Bertschinger and Mozzhorin (2020).

using more realistic test cases that better reflect the current state of economic and financial agent-based modelling.

Recently, there has been a relatively modest increase in the awareness of this point within some groups and increasing emphasis has been placed on addressing these computational difficulties in some circles (Grazzini et al. 2017; Lamperti et al. 2018). A somewhat promising suggestion is the use of surrogate modelling to circumvent the intensive ABM simulation process, with representative examples including Gaussian process interpolation, also known as kriging[32] (Salle and Yildizoglu 2014), and the more general machine learning surrogate approach of Lamperti et al. (2018). Unfortunately, the use of a surrogate model introduces an additional layer of distortion into an already sensitive ABM calibration pipeline. To elaborate, most schemes attempt to learn a computationally efficient surrogate that maps model parameter values to simulated Bayesian posterior densities, SMD objective function values, or other similar quantities using a training set consisting of evaluations of these functions of interest at a number of parameter value combinations. Since most of the aforementioned objective function or posterior density values are already simulation-based approximations, and may, in some cases, be subject to high levels of noise[33], the additional layer of approximation implied by the use of a surrogate model may lead to unacceptably large errors.

As an alternative to the above, recent years have also seen the emergence of cloud computing platforms, such as Amazon Web Services, which provide users with access to computing resources that are essentially rented for a required task, rather than purchased outright, granting access to computing power several orders of magnitude greater than may be possible through traditional on-site means. Therefore, the use of such services may provide researchers with the means to tackle more ambitious calibration problems without the need to resort to surrogate modelling or related approaches. This is generally the approach favoured throughout this thesis when dealing with more complex examples, though Chapter 5 does provide a detailed treatment of the application of surrogate modelling.

1.2 Agent-based Approaches to Financial Market Modelling

While the work carried out in this thesis is broadly applicable to the general class of economic ABMs, much of our focus, particularly in later chapters, is devoted to models attempting to replicate the asset return dynamics observed in financial markets[34]. Although our consideration of this area is primarily motivated by a general interest in the financial domain, there are also a number of practical benefits that make it especially appealing. First and foremost, macroeconomic time series are generally relatively short, with measurements usually being taken at a quarterly or monthly frequency. In contrast to this, daily price time series data is widely available for many of the world's financial markets and intraday datasets are becoming increasingly accessible. Financial markets therefore represent a regime in which data is plentiful and thus provide the ideal starting point for the development and testing of data-intensive statistical methods. Additionally, most of the existing ABM calibration

We discuss these limitations in greater detail in Sections 3.4.2 and 4.1.3.1.

[32]Refer to Barde and van der Hoog (2017) for a first attempt at combining similar methods with the MIC to calibrate a large-scale economic ABM.

[33]Objective function noise and its consequences are explored in detail in Chapter 2.

[34]To be more specific, we focus on the modelling of returns in equity (stock) markets.

literature has also focused on financial applications, implying that the consideration of related examples would allow for greater integration with prior work. It thus follows that a brief review of existing agent-based approaches to financial market modelling is in order.

1.2.1 A Brief Overview of Financial Markets

Given that the ABM paradigm emphasises the direct representation of the actions of and interactions between the microconstituents of a system of interest, a natural starting point would be a discussion of the basic structure of contemporary financial markets. In simplified terms, financial markets consist primarily of agents (traders) who buy or sell assets that may take the form of equities, currency, derivative contracts, and so on. Transactions are typically facilitated through one of two mechanisms: decentralised networks of market makers or a centralised exchange.

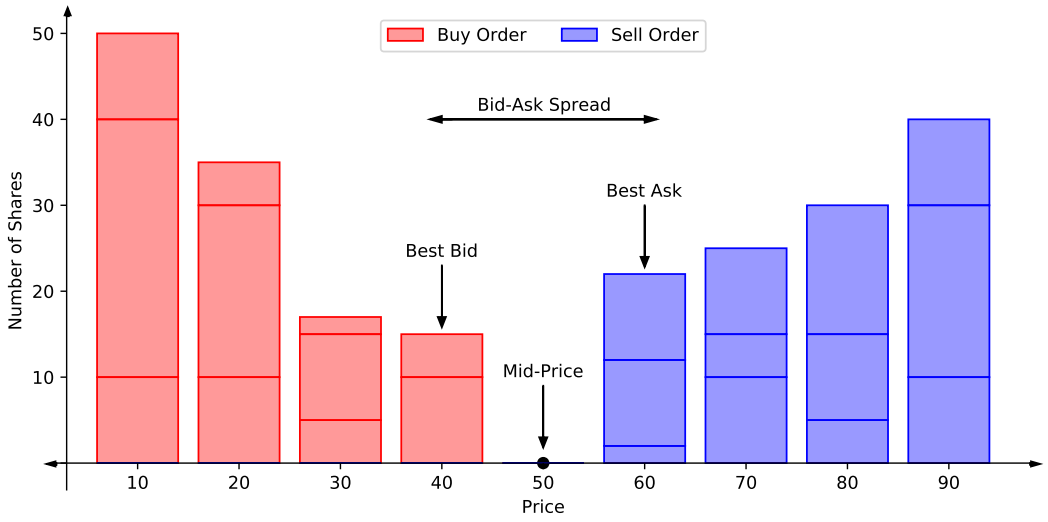


Figure 1.2: An illustration of the basic structure of a typical LOB in a continuous double auction market.

To be more precise, a market maker is an individual or organisation that quotes prices at which it is willing to buy or sell a given asset. In most cases, the price at which a market maker is willing to sell the asset is greater than the price at which it is willing to buy it, with this discrepancy in prices, called the bid-ask spread, being set in order to allow the market maker to both profit and manage risk. Many assets, including bonds, currency, and derivative contracts, are traded in what are referred to as over-the-counter markets, which take the form of a decentralised network of independent market makers who execute trades through private agreements (Attanasi et al. 2016).

In contrast to the above decentralised networks, one may also wish to consider a market with some form of centralised asset exchange, which is typical of equities. Perhaps the most famous formalism of such a paradigm is that of the Walrasian auction (Walras 2013). In such a market, trader agents submit demands to buy or sell the asset to a single auctioneer or market maker that sets the prevailing asset price in order to balance supply and demand,

or, in other words, ensure that the market clears. In practice, modern stock markets are not characterised by the existence of such an auctioneer and most instead make use of a structure known as an order book to facilitate trading via a continuous double auction (Vytelingum et al. 2008). Although seemingly distinct from a Walrasian approach, there are key similarities that should be noted, which will become apparent as we discuss continuous double auction markets in more detail.

To summarise, a continuous double auction market[35] consists of traders who submit orders of two types, limit and market orders, to a centralised exchange during some period of continuous trading[36]. During this period, both new and unexecuted existing orders are matched automatically in real time by an algorithm, which adopts appropriate procedures for handling different order types. A limit order is associated with a specified price, called the limit price, and will only execute should it be possible to buy (sell) the asset at the desired price level or better. Unexecuted limit orders are stored in a structure known as the limit order book (LOB)[37], where they remain until orders with suitable prices arrive and they are executed or until they are cancelled. In many markets, a regime known as price-time priority is adopted, where sell (buy) orders with lower (higher) limit prices are executed first and the oldest order at a set price level is given priority. In contrast, market orders simply execute immediately at the best available price, or more precisely, the price of the current best bid (ask)[38] when attempting to sell (buy) some quantity of the asset.

In essence, the unexecuted sell and buy limit orders constituting the order book may be seen as representing supply and demand respectively and the mid-price, or average of the best bid and ask, may be seen as indicative of the prevailing market price. In this, some parallels may be drawn to the Walrasian auction. Nevertheless, it should be noted that since the current state of the LOB is generally known publicly, trader agents attempt to hide their intentions by placing orders that represent only a fraction of their true demands (Donier and Bouchaud 2016). For this reason, the current state of the order book is likely to provide only limited insights into the true nature of supply and demand and the mid-prices associated with a continuous double auction may therefore behave quite differently to the prices associated with a Walrasian mechanism. Therefore, market clearing assumptions are likely inappropriate for the modelling of real equity markets, at least at shorter or intraday time scales (Kononovicius and Ruseckas 2019). Interestingly, however, it turns out that market clearing assumptions may be more suitable at longer or interday time scales.

1.2.1.1 Inter- and Intraday Regimes

Elaborating on the above, typical trading days in most equity markets follow a similar flow of events: an initial opening auction, an extended period of continuous trading, and finally a closing auction (Gould et al. 2013). Both opening and closing auctions represent an attempt to balance supply and demand and determine fair opening and closing prices respectively. They therefore yield an alternative regime with differing order placement behaviours to those observed during continuous trading and are often viewed as noticeably more Walrasian in nature (Bauwens and Giot 2001).

[35] The interested reader should refer to Gould et al. (2013) for a more detailed introduction to continuous double auction markets and order books.

[36] In most markets, this would refer to a full trading day, with the exception of opening and closing auctions.

[37] We provide a basic illustration of a typical LOB configuration in Figure 1.2.

[38] The best bid (ask) refers to the buy (sell) limit order currently stored in the order book with the highest (lowest) limit price.

From the perspective of the modeller, this implies that the modelling of price movements needs to be approached differently depending on whether an inter- or intraday time scale is being considered. In the interday case, which typically refers to the modelling of daily closing price movements, models incorporating market clearing mechanisms may be justified on the grounds of the auction methodology used to determine closing prices in many real markets. Indeed, there exists at least some empirical evidence that convergence to Walrasian equilibrium is observed at longer time scales (Donier and Bouchaud 2016; Kononovicius and Ruseckas 2019). In the intraday case, however, such assumptions are likely to be far more dubious and market clearing mechanisms would therefore need to be replaced with more appropriate alternatives.

1.2.2 Interday Financial ABMs

Having highlighted the key distinctions between inter- and intraday market regimes above, we now proceed with a brief discussion of existing attempts at employing the agent-based modelling paradigm to replicate interday price movements in equity markets. Most models of this class share the same basic structure in which time proceeds through a sequence of discrete sessions during which trader agents, each of which is following one of a number of possible strategies (essentially sets of simple behavioural rules), submit demands to buy or sell the asset to a lone market maker. This market maker then determines the market price that will prevail at the end of the session by either increasing the previous market price in cases of excess demand or decreasing it in cases of excess supply, thus harking back to a Walrasian paradigm. Agents then adjust their strategies (or switch to other strategy types) in response to the updated market conditions.

Where most of the above models differ is in their specification of the behavioural rules that determine agent demands and strategy switching behaviours. As an example, the Brock and Hommes (1998) model consists of agents who follow one of a number of fundamentalist, trend-following, and contrarian strategies during a given session[39]. At the end of each session, agents compute the profitability of each possible strategy and then determine whether to retain their existing set of behavioural rules or switch to an alternative. In contrast to the Brock and Hommes (1998) model, the model proposed by Alfarano et al. (2008) assumes that agents follow either fundamentalist or noise strategies, but do not switch between them. Rather, noise traders are defined as being in either an optimistic or pessimistic state, with a herding mechanism defining changes in agent sentiment. Franke and Westerhoff (2012) introduce an even more complex framework in which agents switch between chartist and fundamentalist strategies in response to a number of factors, including herding tendencies, past and recent profitability, and predisposition to certain approaches to investing[40].

From the above, it should be relatively clear that a plethora of approaches can be considered when defining the strategies and switching behaviours followed by each agent. It is therefore not surprising that there now exists a very rich literature consisting of a diverse array of approaches. Unfortunately, providing a comprehensive review would be significantly

[39]Fundamentalists are investors who believe that a given stock has an intrinsic or fundamental value that is directly related to macroeconomic, industry-level and company-specific conditions. This is in contrast to chartists (who may be either trend-following or contrarian), who instead use technical methods to identify trends in historical data.

[40]The parameters of ABMs of this class usually correspond to meaningful, interpretable quantities, such as the strength of herding behaviours in a given market or how likely traders are to be predisposed to fundamentalism. This is in contrast to less-interpretable, traditional alternatives, such as ARMA-GARCH and related models.

beyond the scope of this investigation. We do, however, suggest that the interested reader refer to the excellent survey by LeBaron (2006).

1.2.3 Intraday Financial ABMs

Although ABMs of the class described above are elegant, computationally-efficient, and interpretable, they cannot be calibrated to intraday data. This is due to the fact that they all, despite differences in their employed behavioural rules, assume a Walrasian market clearing mechanism. As previously discussed, this assumption cannot be defended at shorter time scales, necessitating an alternative approach. It should come as no surprise that the most viable solution is attempting to directly recreate the order book-driven trading environment observed in real equity markets, leading us to consider the rich literature of LOB modelling approaches.

According to Gould et al. (2013), who provide a detailed survey, LOB models can be classed as belonging to one of three categories: models assuming perfect rationality, zero-intelligence agent (ZIA) models, and ABMs. Clearly, the first of these paradigms is more in line with traditional economic approaches and most models of this class are additionally static and lacking in temporal dynamics. We thus do not consider them explicitly. ZIA models, on the other hand, are dynamic in nature and often model the flow of limit orders, market orders, and order cancellations as stochastic processes[41]. In essence, these models can be interpreted as containing a population of agents all making random trading decisions, rather than following particular strategies or sets of behavioural rules. It is this random trading behaviour that ultimately gives rise to the adoption of the term "zero-intelligence". Interestingly, models of this class, despite being highly-simplified, are capable of replicating a wide range of the statistical regularities observed in real markets (Smith et al. 2003; Ladley 2012). Finally, ABMs, as defined by Gould et al. (2013), involve the specification of interacting, heterogenous agents who submit limit and market orders to a central LOB and follow sets of behavioural rules corresponding to various trading strategies. In this, LOB ABMs can be seen as analogous to the interday ABMs we previously described, but with the central market maker and market clearing mechanisms being replaced by an order book and matching engine. Such models now represent a fairly broad literature consisting of many different strategy specifications[42], all of which are capable of replicating a wide variety of return time series and LOB stylised facts (Panayi et al. 2013; Barde 2016).

In general, we mostly agree with the model categorisation proposed by Gould et al. (2013), but suggest that the nomenclature could be adjusted to better account for some inconsistencies[43]. In particular, we posit that what Gould et al. (2013) refer to as "ABMs" may better be described as "behavioural ABMs", due to their inclusion of behavioural rules and strategies that dictate the flow of orders and cancellations. Additionally, we believe that it would be more natural to consider ZIA models as a subclass of ABMs rather than a separate category, since they still have an agent-based interpretation in which traders simply

[41] The models of Smith et al. (2003) and Cont et al. (2010) are notable examples, though the interested reader may wish to refer to the work of Ladley (2012) for a more comprehensive discussion.

[42] Examples include the model by Chiarella and Iori (2002), in which trader agents follow composite strategies consisting of fundamentalist, chartist, and noise trading components, the investigations of Jacob Leal et al. (2016) and Mandes (2020), which attempt to replicate the interplay between high- and low frequency trading, the work of Navarro and Larralde (2017), which attempts to more accurately represent a number of real-world technical trading strategies, including the moving average oscillator approach, and finally the work of Kononovicius and Ruseckas (2019), which attempts to model herding phenomena in a population of chartist and fundamentalist traders.

[43] See Figure 1.3 for a summary of the classification scheme we employ for financial ABMs.

act randomly as opposed to following purposeful strategies[44]. We believe this refined terminology to be necessary due to the somewhat unclear distinction between what constitutes a ZIA model and a true ABM. As an example, the model of Preis et al. (2006), which Gould et al. (2013) describe as an ABM, is extremely similar to the ZIA model proposed by Smith et al. (2003), with all traders in the former behaving identically to those in the latter, apart from the introduction of basic risk mitigation strategies followed by liquidity providers and other small enhancements[45]. In this, it can be seen that the Preis et al. (2006) model lies somewhere between the behavioural and ZIA extremes, representing what could be classed as a hybrid approach. In practice, we believe that many models are likely to similarly fall somewhere between these two regimes, incorporating some mixture of ZIA and behavioural traits, making a broader definition of ABMs more intuitive[46].

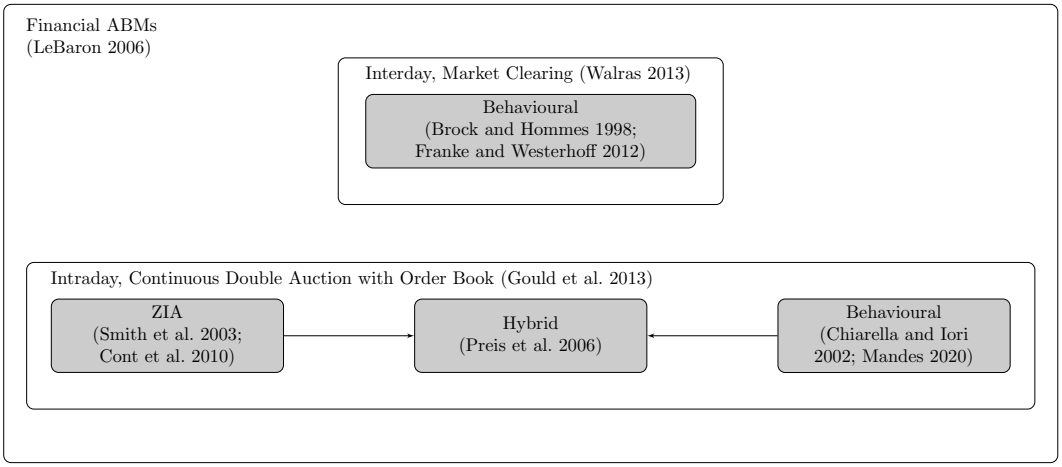


Figure 1.3: An illustration of various financial ABM classes.

In light of the above discussion, it would also be natural to ask whether adding more complex behavioural rules improves the overall explanatory power of ABMs. Put more directly, do ABMs with more complex and realistic trader behaviours produce more realistic asset return dynamics? This remains an open question and has important implications for how LOB modelling is approached[47].

[44]This view is shared by Ladley (2012) and Collver (2017), who both refer to ZIA models as agent-based approaches. Crucially, however, Ladley (2012) indicates that ZIA models should still be differentiated from the more complex behavioural models that are often emphasised in authoritative ABM texts, such as the work of Tesfatsion and Judd (2006). This ultimately leads us to explicitly separate ZIA and behavioural ABMs.

[45]See Section 4.3.2 for a more detailed treatment of both models.

[46]It should be noted that recent years have also seen the emergence of a new class of hybrid models that attempt to replicate the temporal evolution of the order book using reaction-diffusion partial differential equations, which, some might argue, form an entirely new category. We do not consider such models in this thesis, though the interested reader may wish to refer to the work of Mastromatteo et al. (2014) and Dall'Amico et al. (2019) for an introduction.

[47]We explore this question in more detail in Chapter 4.

Chapter 2

A Comparison of Economic Agent-based Model Calibration Methods

In Chapter 1, we provided a comprehensive review of recent developments in the field of ABM calibration and concluded that, although significant progress has been made, particularly in the last three years, this literature still suffers from a number of key weaknesses. Recapitulating briefly, we firstly found it to be overly-compartmentalised, or more specifically, we found that most studies within this research area tend to focus on the proposal of new methods and seldom compare the proposed techniques to other contemporary alternatives. Secondly, we indicated that severe computational limitations have resulted in most techniques only ever being applied to highly-simplified models that are several decades old and no longer a good representation of the current state of economic agent-based modelling, leading to significant doubt regarding the applicability of current methods to the large-scale models that now dominate the literature.

We therefore aim to address the above using a principled, yet practical approach. Specifically, we now proceed to compare a number of prominent ABM calibration methods, both established and novel, through a series of computational experiments involving the calibration of various candidate models in an attempt to determine the respective strengths and weaknesses of each approach. Thereafter, we apply the most promising of the considered calibration techniques to a large-scale ABM of the UK housing market in order to assess the extent to which the performance achieved in the context of simple models is maintained when confronting state-of-the-art ABMs[1].

2.1 Experimental Procedure

In order to effectively compare various calibration techniques, it is necessary to develop a series of tests that quantify the notion of calibration success and result in measures that can be directly compared. One might assume that a natural approach would be the calibration of

[1]This chapter is based on the published article: Platt, D., 2020. A comparison of economic agent-based model calibration methods. J. Econ. Dyn. Control 113, 103859.

a set of candidate models to empirically-observed data using a variety of different approaches and assessing the resulting goodness of fit. Unfortunately, such an approach is likely to be suboptimal for a number of reasons.

Firstly, regardless of the quality and sophistication of a candidate model, it is likely to be misspecified to some extent when compared to the true data-generating process, especially in an economic context. As an example, a given model may be overly-simplified and fail to capture the nuances of the underlying data-generating process, resulting in a poor fit to the data regardless of the merits of each calibration method. Secondly, the notion of goodness of fit is difficult to quantify in the context of ABMs. Indeed, every SMD objective function is an attempt at quantifying it, with each differing in what they consider to be the most important characteristics of the data. We therefore introduce a general approach to calibration method comparison that addresses the above concerns.

2.1.1 Loss Function Construction and Comparison Procedure

We begin by letting $\boldsymbol{X}^{sim}(\boldsymbol{\theta}, T, i)$ be the output of a candidate model, M, for parameter set $\boldsymbol{\theta}$, output length T, and random seed i, where the use of a superscript sim indicates that the quantity being described is derived from or related to simulated rather than real data. Since empirically-observed data is nothing more than a single realisation of the true data-generating process, which may itself be viewed as a model with its own set of parameters, it follows that we may consider $\boldsymbol{X} = \boldsymbol{X}^{sim}(\boldsymbol{\theta}_{true}, T_{emp}, i^*)$ as a proxy for real data to which M may be calibrated.

In this case, we are essentially calibrating a perfectly-specified model to data for which the true set of parameters, $\boldsymbol{\theta}_{true}$, is known. It can be argued that a good calibration method would, in this idealised setting, be able to recover the true set of parameters to some extent and that methods which produce estimates closer to $\boldsymbol{\theta}_{true}$ would be considered superior. It also follows that methods which perform well in this context are far more likely to perform well in the more realistic case of a misspecified model. We therefore define the loss function

$$LS(\boldsymbol{\theta}_{true}, \hat{\boldsymbol{\theta}}) = ||\boldsymbol{\theta}_{true} - \hat{\boldsymbol{\theta}}||_2, \tag{2.1}$$

where $\hat{\boldsymbol{\theta}}$ is the parameter estimate produced by a given calibration method. Therefore, using this approach, we not only address concerns related to misspecified models, but are also able to directly compare calibration methods according to their associated loss function values.

It should be noted that the above construction is made plausible only through our use of pseudo-true data; in settings involving the application of calibration methods to empirical data, it would not be possible to compare the obtained estimates to the true parameter values, as these would be unknown. While we are not suggesting that this is a definitive tool for calibration method comparison, we do believe it to be a principled approach that is able to provide meaningful insights in this context.

Given that we have now defined a metric that allows calibration methods to be directly compared for a given model, M, and true parameter set, $\boldsymbol{\theta}_{true}$, it is relatively straightforward to develop a comprehensive series of comparative tests. We begin by noting that the difficulty of a given test depends on three factors: the complexity of the dynamics produced by M, the number of free parameters in $\boldsymbol{\theta}_{true}$, and the length of the time series data to which M is calibrated, T_{emp}. We must therefore consider a set of models $\boldsymbol{M}$, where each model differs in the complexity and overall nature of its resultant dynamics, and a variety of free parameter

sets of different cardinalities for each model[2]. We can then determine the loss function values associated with each calibration method for the various models and true parameter sets. This will ultimately provide insight into which calibration techniques deliver the best performance and in which situations.

2.1.2 Implemented Models

We now provide descriptions of the implemented models and brief motivations for their consideration in this study. The first four are computationally-inexpensive, appear frequently in the existing calibration literature, and are capable of producing dynamics ranging from very basic to relatively sophisticated. This initial set is selected to identify the most promising among the considered methods and those that perform well in this simplified context will then be applied to a far more computationally-expensive, large-scale ABM of the UK housing market.

The computationally-inexpensive nature of the initial set of models allows us to perform a thorough series of tests involving all of the considered calibration methods in a realistic period of time, after which only methods that are likely to be successful are applied in the more complex setting. Indeed, methods which do not perform well in this simplified context will almost certainly perform poorly when applied to the UK housing market model and we therefore need not waste computational resources on such tests.

2.1.2.1 AR(1) Model

The first model we consider is an autoregressive model of order 1, given by

$$x_{t+1} = a_1 x_t + \epsilon_{t+1}, \tag{2.2}$$

where $\epsilon_t \sim \mathcal{N}(0, 1)$.

It should be apparent that the above is a basic, single parameter model capable of producing a limited range of dynamics and that its calibration should be a trivial exercise. It has been included in order to create a baseline test for which we expect all of the considered calibration methods to perform well.

2.1.2.2 ARMA(2, 2)-ARCH(2) Model

The second model we consider is an ARMA(2, 2) model with ARCH(2) errors, given by

$$\begin{aligned} x_{t+1} &= a_0 + a_1 x_t + a_2 x_{t-1} + b_1 \sigma_t \epsilon_t + b_2 \sigma_{t-1} \epsilon_{t-1} + \sigma_{t+1} \epsilon_{t+1}, \\ \sigma_{t+1}^2 &= c_0 + c_1 \epsilon_t^2 + c_2 \epsilon_{t-1}^2, \end{aligned} \tag{2.3}$$

where $\epsilon_t \sim \mathcal{N}(0, 1)$.

The above is a logical successor to the previously considered AR(1) model, as it is drawn from the same general class, traditional econometric time series models, but is capable of producing more nuanced dynamics and has a parameter space of significantly larger cardinality. Additionally, the calibration of such models is relatively straightforward using

[2]Note that we do not vary the empirical time series length in our experiments, since the effects associated with the number of free parameters and complexity of the model dynamics are generally dominant.

analytical methods. It would thus be worthwhile to assess the performance of simulation-based methods in the context of models known to be amenable to accurate estimation using analytical approaches.

A similar model was considered by Barde (2017) when testing the MIC, though it was employed in the context of model selection rather than calibration.

2.1.2.3 Random Walks with Structural Breaks

The third model we consider is a random walk capable of replicating simple structural breaks, given by

$$x_{t+1} = x_t + d_{t+1} + \epsilon_{t+1}, \tag{2.4}$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$ and

$$d_t, \sigma_t = \begin{cases} d_1, \sigma_1 & t \leq \tau \\ d_2, \sigma_2 & t > \tau. \end{cases} \tag{2.5}$$

Despite the simplicity of the model's equations, its replication of structural breaks and non-stationarity should present a significant challenge to the considered methods.

It should also be noted that the above model was used by Lamperti (2018a) to test the GSL-div, though, in much the same way that the ARMA(2, 2)-ARCH(2) model was employed by Barde (2017) during the testing of the MIC, this was in the context of model selection rather than calibration.

2.1.2.4 Brock and Hommes (1998) Model

The fourth model we consider is the heterogenous agent model proposed by Brock and Hommes (1998), a particular version of which may be expressed as a system of coupled equations:

$$y_{t+1} = \frac{1}{1+r}\left[\sum_{h=1}^{H} n_{h,t+1}(g_h y_t + b_h) + \epsilon_{t+1}\right], \ \epsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{2.6}$$

$$n_{h,t+1} = \frac{\exp(\beta U_{h,t})}{\sum_{h=1}^{H} \exp(\beta U_{h,t})}, \tag{2.7}$$

$$U_{h,t} = (y_t - R y_{t-1})(g_h y_{t-2} + b_h - R y_{t-1}), \tag{2.8}$$

where y_t is the asset price at time t (in deviations from the fundamental value p_t^*), $n_{h,t}$ is the fraction of trader agents following strategy $h \in \{1, 2, \ldots, H\}$ at time t, and $R = 1 + r$.

The above is well-known in the literature as an early example of a class of ABMs that attempt to model the trading of assets in an artificial stock market by simulating the interactions of heterogenous traders that follow various trading strategies[3].

Each strategy, h, has an associated trend following component, g_h, and bias, b_h, both of which are real-valued parameters that are of particular interest in our investigation. In general, the number of strategies, H, may vary from application to application and is typically determined according to the modeller's discretion. We have thus selected $H = 4$ for our exercises. In addition to the aforementioned strategy-specific parameters, the model also includes positive-valued parameters that affect all trader agents, specifically β, which

[3]The interested reader should refer to Brock and Hommes (1998) for a detailed discussion of the model's underlying assumptions and the derivation of the above closed-form solution.

controls the rate at which agents switch between various strategies, and the prevailing market interest rate, r. In most applications, an i.i.d. dividend process is assumed, leading to a constant fundamental value of $p_t^* = p^*$, allowing us to obtain the asset price at time t,

$$x_t = y_t + p^*. \tag{2.9}$$

Despite its relative simplicity, the Brock and Hommes (1998) model is capable of producing a range of sophisticated dynamics, including chaotic behaviours, and is computationally-inexpensive to simulate, unlike most contemporary ABMs. These desirable features have led to it being a popular choice in many ABM calibration exercises throughout the literature, making it a natural inclusion in this investigation.

2.1.2.5 INET Oxford Housing Market Model

The fifth and final model we consider is a large-scale ABM of the UK housing market, which was introduced in the working paper by Baptista et al. (2016). As previously discussed, the consideration of such models in the calibration literature is still very rare, with most recent attempts focusing on far simpler alternatives, such as the Brock and Hommes (1998) model. Since the estimation of large-scale models remains an open problem and is the ultimate goal of current research in the field, attempts to calibrate this model will form a central component of this investigation.

Given its sophistication, however, we are unable to provide a detailed description and instead include an illustrative overview[4]. In summary, the model, which extends the work of Geanakoplos et al. (2012), consists of heterogeneous household agents (buy-to-let investors, owner-occupiers, first-time buyers, and renters) who interact with each other and an aggregate private bank (mortgage provider) and an aggregate central bank (setter of mortgage regulations and the basic interest rate) through a sequence of T sessions. Each session involves, in order:

1. Demographic changes, including the birth, death, and ageing of households
2. The construction of new houses, with random qualities
3. The update of household states, including the collection of income from rent or employment and spending on mortgage payments, rent, and other expenses
4. Decisions on whether to move or purchase/sell properties, followed by the appropriate posting of bids, offers, and mortgage requests:
    - Homeless households decide whether to buy or rent
    - Renters continue to pay until their contract ends, after which they decide whether to purchase a property or continue renting
    - Owner-occupiers decide whether or not to move
    - Buy-to-let investors decide whether to increase or decrease the size of their property portfolios
5. The clearing of the rental and sale markets, both of which are implemented as continuous double auctions

[4]The model source code is publicly available and can be found at `https://github.com/EconomicSL/housing-model`.

6. The recalculation of the mortgage rate and other bank-related actions

The above sequence of events ultimately results in a set of time series, each of length T, including the sale housing price index (HPI), rental HPI, the amount of credit extended to households, the number of sales per month, the distribution of loan-to-value ratios, and the distribution of loan-to-income ratios, any combination of which could potentially be used for calibration purposes.

2.1.3 Implemented Methods

Given that we have now described the overall experimental procedure and selected models in detail, we finally discuss each of the implemented calibration methods, which have been selected such that a number of different niches within the calibration literature have been represented. As a result of the complexity of a number of the considered approaches, this section is primarily illustrative and discusses only the key ideas underpinning each methodology. We therefore recommend that the interested reader refer to the references associated with each method for a more complete discussion.

At this point, it should also be noted that most ABMs of interest are unlikely to be ergodic[5]. Therefore, each evaluation of the objective function or posterior density (depending on whether the method is frequentist or Bayesian) in our investigation involves the comparison of our proxy for real data, $\boldsymbol{X}^{sim}(\boldsymbol{\theta}_{true}, T_{emp}, i^*)$, with an ensemble of R Monte Carlo replications, $\boldsymbol{X}^{sim}(\boldsymbol{\theta}, T_{sim}, i), i = i_0, i_0+1, \ldots, i_0+R-1$, where $i_0 \in \mathbb{N}$ and we assume that $i^* \notin \{i_0, i_0+1, \ldots, i_0+R-1\}$. How these replications are utilised differs from method to method and is elaborated upon when introducing each approach.

2.1.3.1 SMD Objective Functions

Since SMD has historically been the most popular ABM calibration paradigm, it is appropriate that it features prominently in this investigation. Recall that SMD methods involve the construction of an objective function, $f(\boldsymbol{\theta})$, which in our context measures the degree of dissimilarity between pseudo-true data and an ensemble of R model-generated Monte Carlo replications. Naturally, this measure of dissimilarity could be constructed in a multitude of ways, leading us to consider several SMD objective functions, including:

- MSM, which remains a popular choice throughout the ABM calibration and econometric literature, making it an essential inclusion as a benchmark against which to compare newer approaches. While there are a number of different interpretations of the MSM framework and many moment sets that could potentially be considered, we ultimately chose a modified version of the implementation described by Franke (2009), combined with the moment set proposed by Chen and Lux (2018), which includes the variance of the raw series, kurtosis of the raw series, autocorrelation coefficients of the raw series, absolute value series and squared series at lag 1, and the autocorrelation coefficients of the absolute value series and squared series at lag 5. This leads to an objective function defined by

$$f(\boldsymbol{\theta}) = \boldsymbol{g}(\boldsymbol{\theta})^T \boldsymbol{W} \boldsymbol{g}(\boldsymbol{\theta}) \tag{2.10}$$

[5]Refer to Grazzini (2012) for a detailed discussion on ergodicity and its relationship with ABM calibration.

and

$$\boldsymbol{g}(\boldsymbol{\theta}) = \frac{1}{R}\sum_{i=1}^{R}\left[\hat{\boldsymbol{m}}_i^s(\boldsymbol{\theta}) - \hat{\boldsymbol{m}}\right], \tag{2.11}$$

where $\hat{\boldsymbol{m}}_i^s(\boldsymbol{\theta})$ is the vector of (time-averaged) simulated moments estimated from the i-th Monte Carlo replication, $\boldsymbol{X}^{sim}(\boldsymbol{\theta}, T_{sim}, i)$, $\hat{\boldsymbol{m}}$ is the vector of (time-averaged) empirical moments estimated from $\boldsymbol{X}^{sim}(\boldsymbol{\theta}_{true}, T_{emp}, i^*)$, and $\boldsymbol{W}$ is a 7×7 weight matrix calculated as the inverse of the Newey-West estimator of the covariance matrix of the empirical moments. Overall, the idea underpinning this methodology is rather simple: simulated and empirical data are summarised by moment vectors and a dissimilarity measure is constructed based on their distance, with each moment being weighted such that moments estimated with less uncertainty are given greater weight.

- The GSL-div (Lamperti 2018a), a recently-introduced information-theoretic criterion capable of comparing the temporal patterns occurring in different time series. In order to facilitate the calculation of the required information-theoretic quantities, the procedure begins with a discretisation process that converts the simulated and empirically-observed data into series of windows of a given length, l, with each window consisting of sequences of natural numbers between 1 and b and corresponding to a single word in vocabulary $S_{l,b}$. Following this initial step, it then proceeds by calculating the subtracted L-divergence between the word distributions implied by the different datasets (empirically-measured and model-simulated) using frequency-based estimators and finally concludes by weighting and summing the L-divergence estimates for a desired number of window lengths, L, and adding a bias correction term. This leads to an objective function defined by

$$\begin{aligned} f(\boldsymbol{\theta}) = \sum_{l=1}^{L} w_l \mathbb{E}\left[-2\sum_{s\in S_{l,b}} m_l^{\boldsymbol{\theta}}(s)\log_{a_{l,b}} m_l^{\boldsymbol{\theta}}(s) + \sum_{s\in S_{l,b}} \hat{p}_l^{\boldsymbol{\theta}}(s)\log_{a_{l,b}} \hat{p}_l^{\boldsymbol{\theta}}(s)\right] \\ + \sum_{l=1}^{L} w_l \left(\frac{B^{m_l^{\boldsymbol{\theta}}} - 1}{4(T_{sim} - l + 1)} - \frac{B^{\hat{p}_l^{\boldsymbol{\theta}}} - 1}{2(T_{sim} - l + 1)}\right), \end{aligned} \tag{2.12}$$

where $m_l^{\boldsymbol{\theta}}(s) = \frac{\hat{p}_l^{\boldsymbol{\theta}}(s) + \hat{p}_l(s)}{2}$, $\hat{p}_l(s)$ is the frequency of word s in the discretised empirical series, $\hat{p}_l^{\boldsymbol{\theta}}(s)$ is the frequency of word s in a discretised simulated series, w_l are positive weights such that $\sum_l w_l = 1$, $B^{m_l^{\boldsymbol{\theta}}}$ and $B^{\hat{p}_l^{\boldsymbol{\theta}}}$ are the cardinalities of the supports of $m_l^{\boldsymbol{\theta}}$ and $\hat{p}_l^{\boldsymbol{\theta}}$ respectively, $a_{l,b}$ is the cardinality of $S_{l,b}$, and the expectation is taken with respect to the ensemble of Monte Carlo replications. Unlike MSM, the GSL-div is capable of comparing time series data directly and does not require any of the datasets to be summarised as a set of moments, making it a compelling alternative.

- The MIC (Barde 2017), which despite also being an information-theoretic criterion and therefore direct competitor to the GSL-div, adopts a fundamentally different approach to achieve its goals. Much like the GSL-div, it begins with a discretisation process. In this case, however, each time series observation is converted to a series of r binary digits. This facilitates the application of an algorithm known as context tree weighting (CTW) (Willems et al. 1995) to the ensemble of model-simulated Monte Carlo replications, which results in the construction of an N-th order Markov process representing the probability of observing a 1 after particular sequences of bits in the

current observation and the preceding L observations. The obtained transition probabilities are then used to calculate binary log scores for each discretised observation in the empirically-observed data, which are then summed to produce an estimate of the cross entropy between the distributions implied by the simulated and empirically-observed data. This leads to an objective function defined by

$$f(\boldsymbol{\theta}) = \sum_{t} \left[\lambda \left(\tilde{\boldsymbol{x}}_t | \tilde{\boldsymbol{x}}_{t-L}^{t-1}, \boldsymbol{\theta} \right) - \epsilon \left(\tilde{\boldsymbol{x}}_t | \tilde{\boldsymbol{x}}_{t-L}^{t-1}, \boldsymbol{\theta} \right) \right] \tag{2.13}$$

and

$$\lambda \left(\tilde{\boldsymbol{x}}_t | \tilde{\boldsymbol{x}}_{t-L}^{t-1}, \boldsymbol{\theta} \right) = - \sum_{k=1}^{r} \left[\tilde{\boldsymbol{x}}_t\{k\} \log_2 \hat{p}^{\boldsymbol{\theta}}(k) + (1 - \tilde{\boldsymbol{x}}_t\{k\}) \log_2 (1 - \hat{p}^{\boldsymbol{\theta}}(k)) \right], \tag{2.14}$$

where $\tilde{\boldsymbol{x}}_t$ represents discretised observation t, $\tilde{\boldsymbol{x}}_{t-L}^{t-1}$ represents discretised observations $t-L$ to $t-1$, $\tilde{\boldsymbol{x}}_t\{k\}$ represents the k-th bit of discretised observation t, $\hat{p}^{\boldsymbol{\theta}}(k)$ represents the probability (as determined by the CTW algorithm) of $\tilde{\boldsymbol{x}}_t\{k\}$ being 1 conditional on the bits directly preceding it, and $\epsilon \left(\tilde{\boldsymbol{x}}_t | \tilde{\boldsymbol{x}}_{t-L}^{t-1}, \boldsymbol{\theta} \right)$ is a bias correction term. Much like the GSL-div, the MIC does not require the selection of an arbitrary set of moments and has therefore attracted interest as a potential solution to the ABM calibration problem.

Before introducing the optimisation algorithms that we employ to minimise the above objective functions, there is an important nuance that should be pointed out. Specifically, both the GSL-div and MIC, though used in the context of calibration in this investigation, were originally proposed as model selection tools. In such a framework, simulated data generated by various models is compared to a single empirical equivalent and the model resulting in the lowest criterion value is deemed to be the most appropriate. Given that either criterion simply defines some notion of distance between simulated and empirical data, however, one can assume that instead of representing different models, each candidate is simply the same model initialised with a different set of parameter values, making either criterion useable for calibration. The use of information-theoretic criteria for such purposes has basis in the literature, with Lamperti (2018a) arguing for the use of the GSL-div as a calibration tool and Lamperti (2018b) providing a first attempt at employing it for parameter space exploration. Additionally, Barde and van der Hoog (2017) provide attempts at related exercises when applying the MIC to the well-known Eurace model (Cincotti et al. 2010).

2.1.3.2 Optimisation Algorithms

While the construction of the objective function is the central focus of most SMD approaches, a second and equally-important concern is the choice of optimisation method. In general, the resulting optimisation problem is quite difficult and needs to be approached with caution.

In more detail, the simulation-based nature of the considered objective functions implies that gradient-based or related approaches that require analytical expressions for the value of the objective function at a given point are not applicable. This ultimately forces us to consider heuristic methods, which often produce solutions with properties that are not well-understood and do not guarantee convergence to a global minimum (Gilli and Winker 2003; Fabretti 2013). Further, and even more importantly, such methods may require a significant number of objective function evaluations to be performed in order for convergence to be observed, a potential concern when each evaluation is expensive.

Given that there are no best practices in the literature, we will consider two contemporary heuristics, namely:

- Particle swarm optimisation, a class of evolutionary algorithms that mimic the flocking and swarming behaviours of organisms in ecological systems. The most basic variant involves a set of $N_{particles}$ particles, each of which is assigned random initial position and velocity vectors in the parameter space. The algorithm then proceeds for a fixed number of iterations or generations, $T_{generations}$, in which each particle updates its position by moving in the direction implied by its velocity vector, evaluates the objective function at its new position, and updates its velocity. The velocity vector update is a key feature of the algorithm and is constructed such that a balance between movement towards the best candidate parameter set previously found by the swarm as a whole and the best candidate parameter set previously found by the particle itself is achieved. This balance between local and global information (along with stochasticity in the velocity update itself) provides a modest degree of robustness to convergence to local minima, since any particle may at any time move contrary to the overall direction of the swarm and thus potentially aid the swarm as a whole in escaping a local minimum. Algorithms of this nature perform competitively in practice and are now standard in the optimisation literature, with an accessible overview provided by Kaveh (2017).

- The approach of Knysh and Korkolis (2016), a surrogate model method based on the work of Regis and Shoemaker (2005) and designed for the optimisation of expensive blackbox functions[6]. In brief, the algorithm begins by efficiently sampling N_{LHS} points of the parameter space using latin hypercube sampling and proceeds to evaluate the objective function at the sampled points. Thereafter, a computationally-inexpensive approximation to the objective function is constructed using radial basis functions, which can then be minimised to obtain an initial solution[7]. This initial solution is then gradually improved through the consideration of N_{CORS} further evaluations of the original objective function in an iterative procedure known as the CORS algorithm. This iterative process ultimately allows the optimisation routine to investigate areas of the parameter space that were previously poorly-explored and helps address issues of convergence to local minima. In contrast to particle swarm optimisation, the method proposed by Knysh and Korkolis (2016) could be seen as a more intensive approach; it would thus be interesting to compare their performance for our considered set of problems.

2.1.3.3 Likelihood Approximation and Bayesian Estimation

As previously suggested, the consideration of Bayesian inference in the ABM calibration literature is very much a recent development. For this reason, we aim to compare the approach of Grazzini et al. (2017), which provides a first attempt at constructing a simple likelihood function and employing it within a Bayesian framework, to the dominant SMD paradigm.

In essence, the approach assumes that observations in each of the considered datasets (model-simulated or empirically-observed) are i.i.d. random samples from unconditional

[6]Source code is available at `https://github.com/paulknysh/blackbox`.

[7]Note that this is similar to the kriging approach of Salle and Yildizoglu (2014), discussed in Section 1.1.4.2.

distributions that characterise observations in each dataset. Under this assumption, KDE is used to construct a density function that characterises the distribution of observations in the simulated data for a given set of parameter values, $\boldsymbol{\theta}$, which can then be used to determine the likelihood of the empirically-observed data given this parameter set as

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{t=1}^{T} \tilde{f}(\boldsymbol{x}_t|\boldsymbol{\theta}), \tag{2.15}$$

where $\boldsymbol{X} = \boldsymbol{X}^{sim}(\boldsymbol{\theta}_{true}, T_{emp}, i^*) = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{T_{emp}}\right]$ is the empirical data and function $\tilde{f}$ is determined by applying KDE to data simulated by the candidate model when initialised using parameter set $\boldsymbol{\theta}$, as described above. As in the case of SMD methods, we generate an ensemble of Monte Carlo replications for each candidate parameter set, $\boldsymbol{\theta}$, rather than a single realisation, and combine the samples from each of the replications into a single, larger sample before applying KDE.

Following the construction of the likelihood, the approach proceeds by applying Bayes' theorem, which ultimately results in a posterior distribution, $p(\boldsymbol{\theta}|\boldsymbol{X})$, that is amenable to sampling using a Metropolis-Hastings or related Markov chain Monte Carlo algorithm. In our context, we require a point estimate rather than a complete distribution in order to apply our loss function. In most cases and particularly for parameters taking on continuous values, the mean or median are preferred, since the mode may be an atypical point (Murphy 2012). In our case, we consider the posterior mean, since it corresponds to the optimal estimate for the squared error loss and thus our chosen loss function[8].

2.1.3.4 Sampling Algorithm

Once we are able to evaluate the likelihood of the empirically-observed data for a particular set of parameter values, we require a method to draw samples from the corresponding posterior distribution, with Grazzini et al. (2017) proposing a random walk Metropolis-Hastings algorithm. While we found this to be effective for most of the problems considered in earlier versions of our experiments, this typically required a careful adjustment of the step size in order to obtain an appropriate acceptance rate, which inevitably necessitated the repetition of some of the experiments. For more complex models, such as the housing market model discussed in Section 2.1.2, this proved to be highly-inefficient due to the computational costs associated with each experiment.

In response, we ultimately opted to use the adaptive Metropolis-Hastings algorithm proposed by Griffin and Walker (2013) in all experiments in which Bayesian estimation was attempted, leading to significant improvements in the robustness of the sampling process. Overall, our discussion here is mainly illustrative and positioned in the context of our investigation and the interested reader should therefore refer to the original contribution by Griffin and Walker (2013) for theoretical justifications and a more general discussion[9].

In essence, the approach is centred on the idea of maintaining a set of N samples, $\boldsymbol{\theta}_s = \left\{\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right\}$, $s = 1, 2, \ldots, S$, that is updated for a desired number of iterations. Initially, the set consists of samples drawn uniformly from the space of feasible parameter values, $\boldsymbol{\Theta}$, but eventually converges to be distributed according to $p(\boldsymbol{\theta}|\boldsymbol{X})$. This is achieved through the construction of an adaptive proposal distribution that is dependent on the current samples, $\boldsymbol{\theta}_s$, which can be summarised algorithmically as follows:

[8] This is motivated further in Section 2.2.1.5.

[9] We also apply a minor modification to the algorithm, which is discussed in Appendix 2.B.

1. Sample $\boldsymbol{z}$ according to $\tilde{p}\left(\boldsymbol{z}\middle|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)$, which is determined by applying KDE to $\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N)}$.

2. Propose the switch of $\boldsymbol{z}$ with $\boldsymbol{\theta}_s^{(n)}$, where n is chosen uniformly from $\{1, 2, \ldots, N\}$.

3. Accept the switch with probability

$$\alpha = \min\left\{1, \frac{p\left(\boldsymbol{z}\middle|\boldsymbol{X}\right)\tilde{p}\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(n-1)}, \boldsymbol{z}, \boldsymbol{\theta}_s^{(n+1)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)}{p\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{X}\right)\tilde{p}\left(\boldsymbol{z}|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)}\right\}. \tag{2.16}$$

4. If accepted, set $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}_s$ with $\boldsymbol{\theta}_s^{(n)}$ replaced by $\boldsymbol{z}$, otherwise simply set $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}_s$.

Repeating the above for S iterations, we obtain a sequence of S sample sets, each of size N, that can be used to compute expectations of the form

$$\mathbb{E}\left[g(\boldsymbol{\theta})\right] = \frac{1}{NS}\sum_{s=1}^{S}\sum_{n=1}^{N} g\left(\boldsymbol{\theta}_s^{(n)}\right), \tag{2.17}$$

allowing us to estimate the posterior mean as required.

Ultimately, this has become our MCMC algorithm of choice for two main reasons:

1. The number of iterations required to reach convergence in random walk Metropolis-Hastings algorithms depends significantly on the initialisation of the algorithm. If, for example, the initial candidate parameter set has a particularly low posterior density, it could take a substantial period of time before convergence is observed. Since the algorithm proposed by Griffin and Walker (2013) is initialised using a sample of points from a number of areas of the parameter space, this problem is less pronounced.

2. Most random walk Metropolis-Hastings algorithms require careful tuning of the proposal distribution, usually with the aim of obtaining an acceptance rate of roughly 25%, in order to ensure a good balance between local exploration of high density areas of the parameter space and global coverage of the parameter space as a whole (Robert and Casella 2010). This can be difficult to achieve in practice, making an adaptive approach that determines the proposal distribution automatically particularly appealing.

2.2 Results and Discussion

In Section 2.1, we described a procedure for comparing the effectiveness of various ABM calibration techniques using computational experiments. We now present the results of these experiments and discuss their implications.

2.2.1 Simple Time Series Models

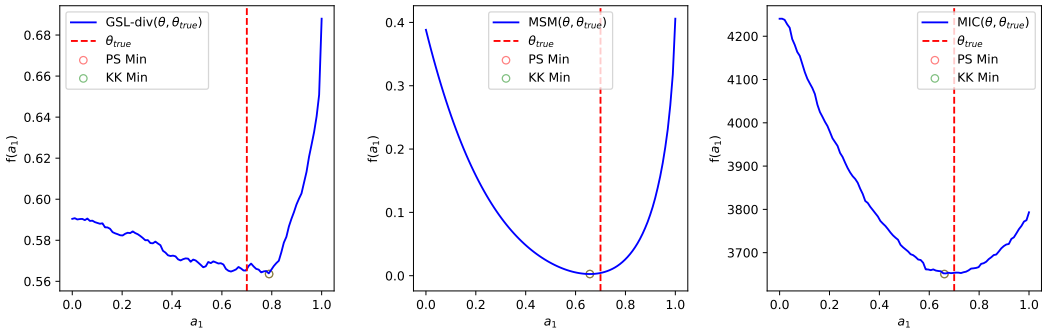
The first series of tests involves the application of all of the implemented calibration techniques to a set of simple time series models. While the main elements of the experimental procedure have been outlined in Section 2.1, we have also provided a more detailed overview of the technical details of these experiments in Appendix 2.A, including discussions related to dataset construction and the setting of the hyperparameters for each calibration method.

2.2.1.1 AR(1) Model

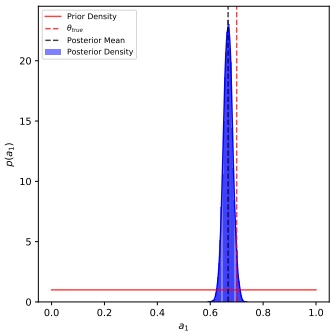
Table 2.1: Calibration Results for the AR(1) Model

	a_1	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	0.7	–
GSL-div/PS	0.7893	0.0893
GSL-div/KK	0.7894	0.0894
MSM/PS	0.6573	0.0427
MSM/KK	0.6572	0.0428
MIC/PS	0.6615	0.0385
MIC/KK	0.6601	0.0399
BE	0.6676	0.0324

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016).



(a) SMD objective function curves for parameter a_1.



(b) Posterior distribution of parameter a_1.

Figure 2.1: A graphical illustration of the calibration results obtained for the AR(1) model, with PS referring to particle swarm optimisation and KK referring to the approach of Knysh and Korkolis (2016).

We now proceed with the presentation of the results of our proposed comparative experiments, beginning with those obtained when attempting to calibrate the AR(1) model. Since the model has only a single parameter, $a_1 \in [0, 1]$, and forms part of a baseline test that all of the considered methods are expected to perform well in, our discussion will be relatively brief.

Referring to Figure 2.1a, where we plot the objective function curves for each of the considered SMD methods and the corresponding minima found by each optimisation algorithm, we find that, as expected, each method produces estimates close to the true parameter value with little difficulty. Similarly, when referring to Figure 2.1b, where we present the posterior distribution of a_1 obtained using Bayesian estimation, we find that the posterior mean (our selected point estimate) is indeed also close to the true parameter value.

More detailed results, including the loss function values associated with each point estimate, are presented in Table 2.1, with the loss function suggesting that Bayesian estimation delivers the best performance for the simple AR(1) model, followed by the MIC, MSM, and finally the GSL-div.

2.2.1.2 ARMA$(2, 2)$-ARCH(2) Model

The next model to be calibrated, an ARMA$(2, 2)$-ARCH(2) model, has a larger parameter space and is capable of producing more complex dynamics than the AR(1) model. It therefore warrants a more comprehensive investigation, leading us to consider two free parameter sets[10], $[a_0, a_1]$ and $[b_1, b_2, c_0, c_1, c_2]$, where all parameters are assumed to lie in the interval $[0, 1]$, with the exception of a_1, which we assume lies in the interval $[0, 0.8]$[11].

As is evident in Figure 2.2, where we plot the objective function surfaces associated with the first free parameter set, differences in the relative performances of the GSL-div, MSM and MIC have become more pronounced when attempting to calibrate this more sophisticated model. While MSM and the MIC produce reasonable parameter estimates, the GSL-div appears to have performed relatively poorly, yielding an estimate for a_0 that is significantly different from its true value. A more thorough inspection of the objective function surface reveals that changes in the value of a_0 appear to have a limited effect on the GSL-div. Since a_0 is simply an additive constant, this likely stems from the fact that the GSL-div is constructed[12] in such a way that it focuses only on relative temporal patterns and is generally agnostic to the magnitude of observations in a given time series, a weakness that is clearly not shared by the MIC and MSM. It should also be noted that the objective function surfaces demonstrate varying degrees of noise. While the MSM objective function surface is very smooth, both the GSL-div and MIC demonstrate a degree of noisiness.

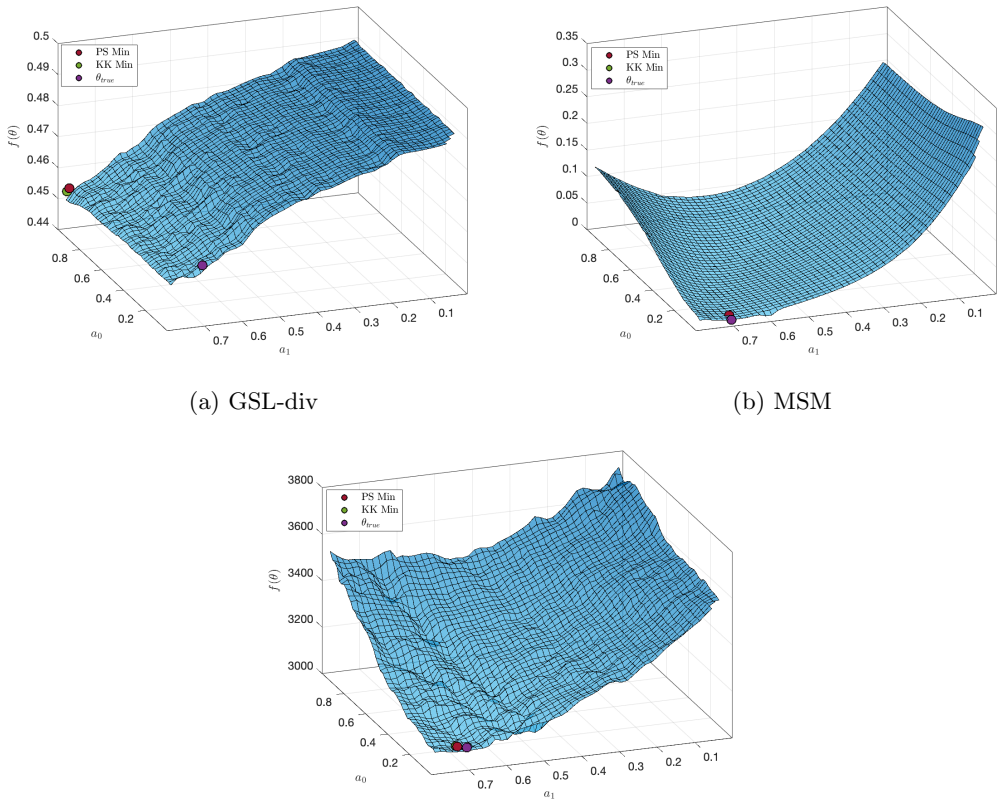
In the case of Bayesian estimation, we see that the method of Grazzini et al. (2017) once again performs well, with the means of the posterior distributions of a_0 and a_1, shown in Figure 2.3, producing reasonable estimates of the true parameter values. The loss function values presented in Table 2.2 also suggest that Bayesian estimation is again the best per-

[10] In a number of experiments throughout this investigation, the free parameter sets represent only a subset of the candidate model's parameters. In such cases, the remaining model parameters are fixed to their true values (those used to generate the pseudo-true data), which are presented in Table 2.11. This allows us to consider problems of a more tractable size, while still employing models that produce relatively complex dynamics.

[11] Values of $a_1 > 0.8$ can lead to the model becoming non-stationary.

[12] The GSL-div uses different discretisation bounds for each series it compares, specifically the minimum and maximum (or 1st and 99th percentiles) of each series, in contrast to the MIC, which uses common discretisation bounds throughout.

forming method, followed by the MIC, MSM and finally the GSL-div, as was the case for the AR(1) model.



(a) GSL-div

(b) MSM

(c) MIC

Figure 2.2: SMD objective function surfaces for free parameter set 1 of the ARMA$(2,2)$-ARCH(2) model, with PS referring to particle swarm optimisation and KK referring to the approach of Knysh and Korkolis (2016).

At this point, it is natural to ask whether similar behaviours are observed for free parameter sets of greater cardinality. Referring to Figure 2.4, it is apparent that the performance of Bayesian estimation is indeed maintained in the more challenging context of 5 free parameters, with the resultant estimates again being reasonably close to the true parameter values. Nevertheless, the uncertainty of estimation is greater than in the previously presented cases, as suggested by the increased variance of the posterior distributions obtained for the second free parameter set. This is to be expected, however, since the number of parameter combinations that produce a reasonable fit to a given dataset is likely to increase with the number of free parameters.

The performance of the considered SMD methods requires a more nuanced discussion. Referring to Table 2.3, we see that the GSL-div and MIC estimates associated with different optimisation algorithms no longer closely match. This is in contrast to MSM, where there is

still close agreement, and suggests that one or both of the considered optimisation algorithms may be converging to local minima when being applied to the MIC or GSL-div. A more thorough analysis reveals that despite their differences, the estimates produced by either optimisation algorithm are generally in the same area of the parameter space[13], suggesting that while both optimisation algorithms are able to reliably narrow down the parameter space to an approximate region surrounding the global minimum, they differ in their ability to effectively probe this region.

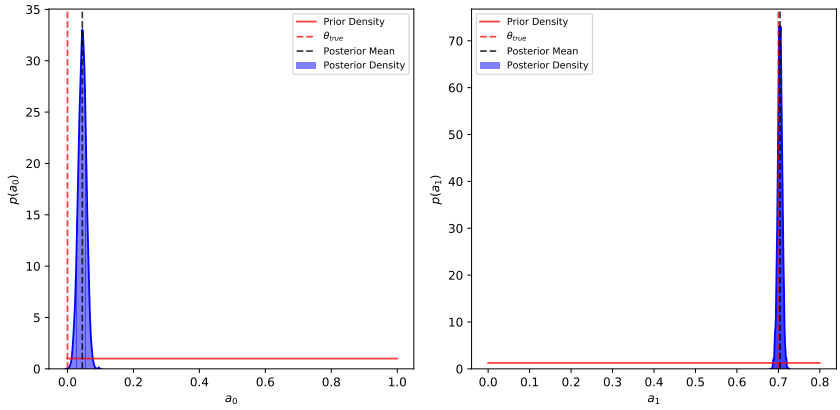


Figure 2.3: Marginal posterior distributions for free parameter set 1 of the ARMA$(2,2)$-ARCH(2) model.

Table 2.2: Calibration Results for Free Parameter Set 1 of the ARMA$(2,2)$-ARCH(2) Model

	a_0	a_1	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	0	0.7	–
GSL-div/PS	1.0	0.8	1.0050
GSL-div/KK	1.0	0.8	1.0050
MSM/PS	0.0649	0.6927	0.0653
MSM/KK	0.0650	0.6926	0.0654
MIC/PS	0.0458	0.7179	0.0492
MIC/KK	0.0506	0.7191	0.0541
BE	0.0449	0.7039	0.0451

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016).

Referring to Figure 2.5, where we provide cross sections of the objective functions, we see that the overall slope of the objective function surfaces is generally far more pronounced further away from the global minimum, with a flattening-out being observed once the global minimum is approached. In cases where the objective function is smooth, such as MSM, this

[13]For example, the estimates associated with different optimisation algorithms do not differ by more than 15% of the total explored parameter range for almost all of the considered free parameters.

is not of much concern and the optimisation algorithms are still able to perform well, since there are clear differences between the objective function values in the region surrounding the global minimum, even if they may be quite subtle. In the case of noisy objective functions (like the GSL-div or MIC), however, convergence to local minima becomes far more likely, since the effect of the objective function noise may become more significant than the effect of the parameters themselves. Therefore, even if a given objective function were to produce an accurate measure of distance between the datasets, if this measure is very noisy, one may inevitably have to settle for estimates that are likely in the same region of the parameter space as the global minimum, but not necessarily the global minimum itself, with the problem likely to become more pronounced in higher dimensions, regardless of the optimisation approach employed.

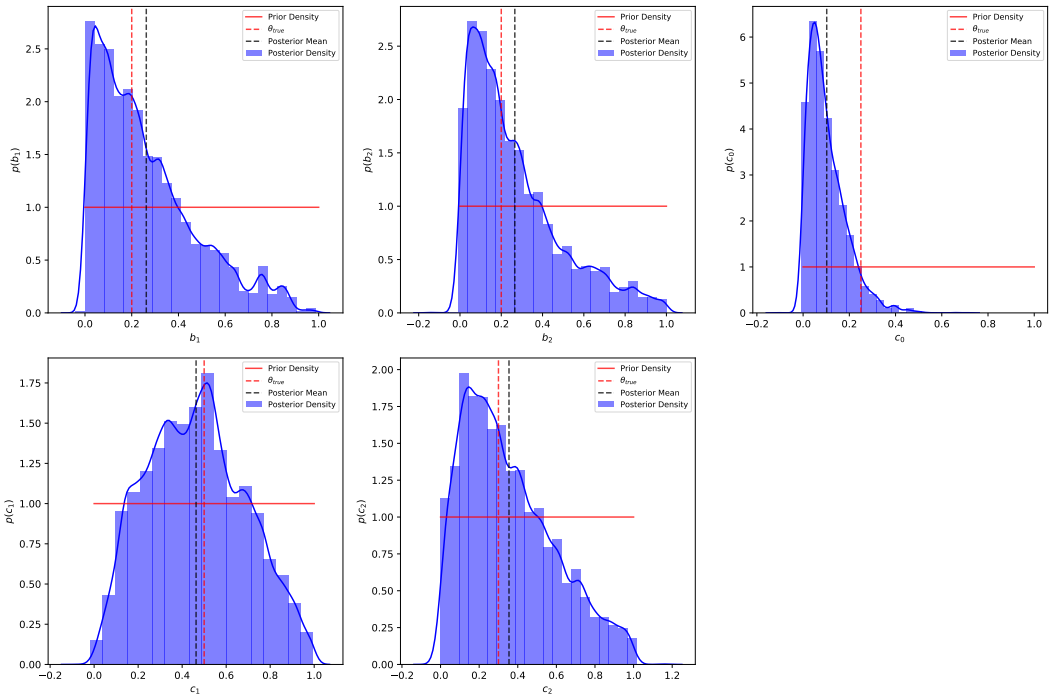


Figure 2.4: Marginal posterior distributions for free parameter set 2 of the ARMA$(2, 2)$-ARCH(2) model.

Along these lines, we find the approach of Knysh and Korkolis (2016) to be preferable to particle swarm optimisation, since it generally finds parameter sets that return lower objective function values in higher dimensions. This is likely due to its initial sampling of the parameter space using a latin hypercube, which allows it to more easily find the region surrounding the global minimum and then iteratively explore this area in more detail.

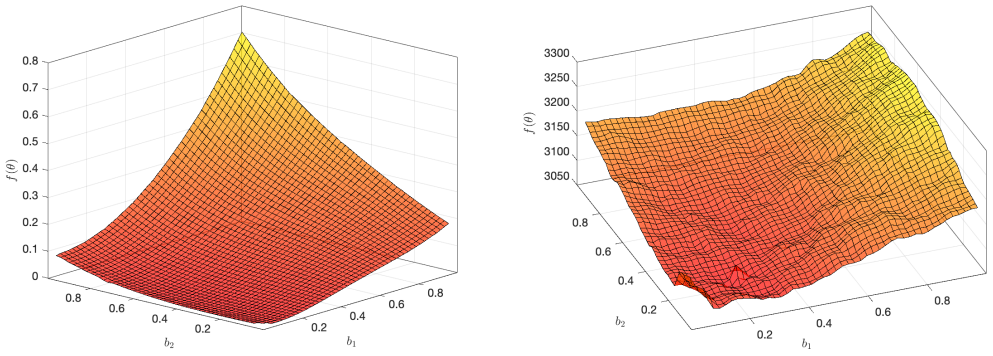
Based on the above discussions and the loss function values presented in Table 2.3, it is clear that the performances of the GSL-div and MSM have significantly decreased for this larger free parameter set, with the associated estimates being very different from the true parameter values, regardless of the optimisation algorithm employed. In contrast to this, we see that the MIC has fared better, despite being difficult to optimise, by producing

Table 2.3: Calibration Results for Free Parameter Set 2 of the ARMA(2, 2)-ARCH(2) Model

	b_1	b_2	c_0	c_1	c_2	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	0.2	0.2	0.25	0.5	0.3	–
GSL-div/PS	0.1310	0.0558	1	0	0	0.9634
GSL-div/KK	0.0734	0.0804	0.7941	0.0137	0	0.8080
MSM/PS	0.0836	0.2070	0	0.1286	1	0.8391
MSM/KK	0.0872	0.2243	0	0.1521	1	0.8288
MIC/PS	0.0944	0.2420	0.3171	0.8199	0.0505	0.4266
MIC/KK	0.2263	0.1583	0.3018	0.6469	0.1887	0.1977
BE	0.2624	0.2656	0.1022	0.4633	0.3546	0.1854

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016).

a reasonable estimate when minimised using the approach of Knysh and Korkolis (2016), indicating that it is a more accurate measure of distance than the GSL-div or MSM in this case.



(a) MSM

(b) MIC

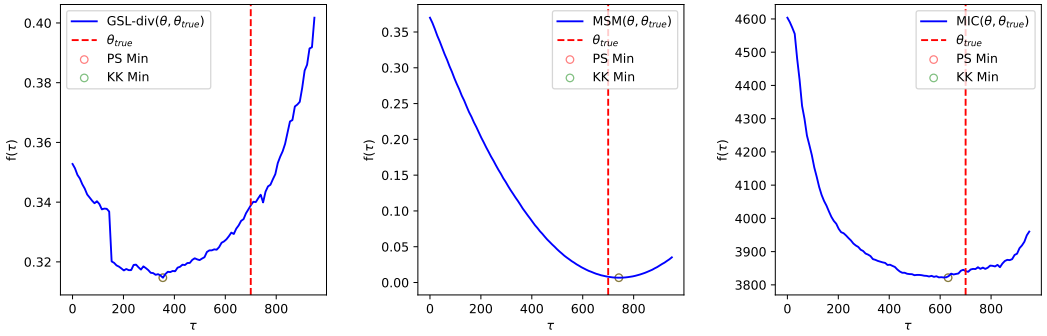
Figure 2.5: SMD objective function cross sections for free parameter set 2 of the ARMA(2, 2)-ARCH(2) model. The values of c_0, c_1, and c_2 have been fixed according to MSM/KK and MIC/KK (see Table 2.3) respectively.

Overall, the ranking of the calibration methodologies according to their loss function values is unchanged, with Bayesian estimation again being the best performing method.

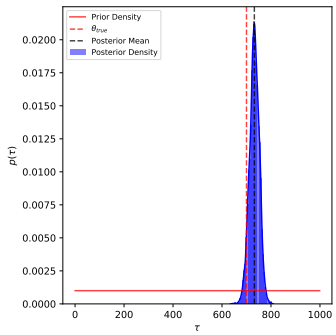
2.2.1.3 Random Walks with Structural Breaks

While the ARMA(2, 2)-ARCH(2) model is indeed capable of producing significantly more nuanced dynamics than the AR(1) model, it is still unable to replicate a range of behaviours observed in both real economic data and the outputs of large-scale ABMs. Among these phenomena, structural breaks, or sudden and dramatic deviations from prior temporal trends,

are of particular interest. This leads us to investigate the extent to which a random walk model capable of producing simple structural breaks can be successfully calibrated using the considered methods.



(a) SMD objective function curves for parameter τ.



(b) Posterior distribution of parameter τ.

Figure 2.6: A graphical illustration of the calibration results obtained for parameter τ of the random walk model, with PS referring to particle swarm optimisation and KK referring to the approach of Knysh and Korkolis (2016).

In this simplified context, a successful method would not only be able to calibrate parameters which determine the pre- and post-break dynamics, but should also be able to identify the point at which the structural break occurs, even if not precisely. We therefore consider two free parameter sets, $[\tau] \in [0, 1000]$ and $[d_1, d_2], d_i \in [0, 1]$, which correspond to each of the aforementioned aspects.

Beginning with Figure 2.6a, which presents the SMD objective function curves resulting from attempts to determine the location of the structural break, we see that both MSM and the MIC are capable of inferring the correct location to some extent, although the estimate associated with MSM is slightly closer to the true value. The GSL-div, on the other hand, delivers far less compelling results. Moving away from SMD methods, Figure 2.6b illustrates the ability of Bayesian estimation to reliably estimate τ, with the loss function values presented in Table 2.4 suggesting that Bayesian estimation again delivers the best performance among the considered methods.

At this point, one can identify the emergence of a persistent trend that appears through-

out the preceding experiments. In particular, notice that Bayesian estimation always delivers the best calibration performance (as measured by the loss function) and has, in all of the cases considered thus far, been relatively successful in recovering the parameters used to generate the pseudo-true data. In contrast to this, SMD methods are less consistent in their performance, with some methods performing well in certain contexts and poorly in others. In most cases, however, it would appear that MSM and the MIC perform better than the GSL-div.

Table 2.4: Calibration Results for Parameter τ of the Random Walk Model

	τ	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	700	–
GSL-div/PS	355	345
GSL-div/KK	355	345
MSM/PS	742	42
MSM/KK	742	42
MIC/PS	631	69
MIC/KK	631	69
BE	732	32

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016).

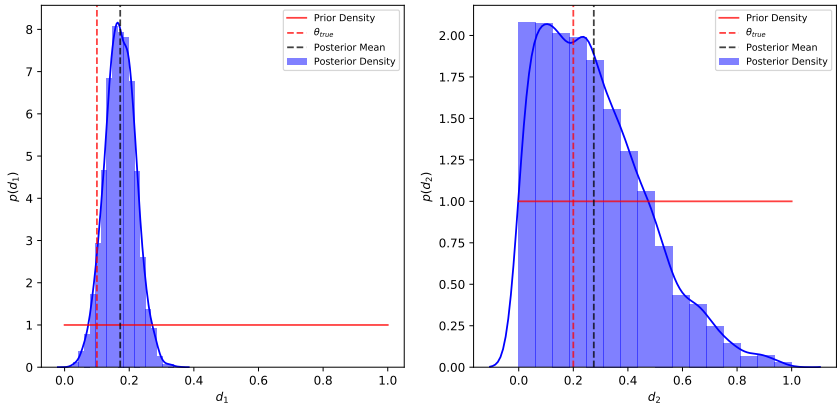
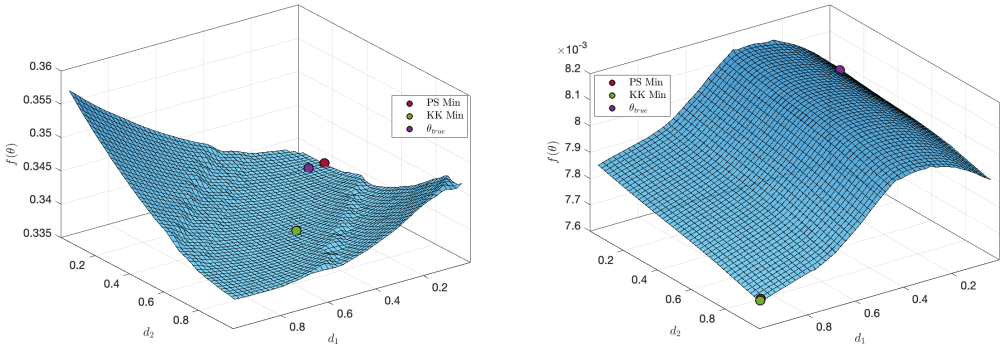


Figure 2.7: Marginal posterior distributions for free parameter set 2 of the random walk model.

Unsurprisingly, Figures 2.7 and 2.8 demonstrate that this trend largely persists in the case of the second free parameter set. Notice, however, that the loss function values presented in Table 2.5 suggest that the GSL-div estimate obtained using particle swarm optimisation is superior when compared to the posterior mean associated with Bayesian estimation, while, paradoxically, the GSL-div estimate obtained using the method of Knysh and Korkolis (2016) is significantly worse. In this case, the disagreement between the estimates associated with

the different optimisation algorithms is not due to objective function noise, as was the case for the second free parameter set of the ARMA$(2,2)$-ARCH(2) model, but is rather due to the inability of the GSL-div to distinguish between changes in the model dynamics induced by changes in d_1 and d_2. In more detail, Figure 2.8 reveals that the GSL-div objective function surface appears to be flat for most of the parameter space. Further, the objective function values associated with the true parameter values and both GSL-div estimates are virtually identical[14]. Therefore, a different estimate is likely to be returned each time, regardless of the employed optimisation method, as we have seen here, and this can generally be taken as evidence that the GSL-div is not effective in this case.



(a) GSL-div

(b) MSM

(c) MIC

Figure 2.8: SMD objective function surfaces for free parameter set 2 of the random walk model, with PS referring to particle swarm optimisation and KK referring to the approach of Knysh and Korkolis (2016).

2.2.1.4 Brock and Hommes (1998) Model

The final simple model to be investigated is that of Brock and Hommes (1998), considered to be a classic example of the use of heterogenous agents to model financial time series.

[14] $f(\boldsymbol{\theta}_{true}) = f(\hat{\boldsymbol{\theta}}_{\text{GSL-div/PS}}) = f(\hat{\boldsymbol{\theta}}_{\text{GSL-div/KK}}) = 0.3388$.

Given the model's incorporation of a number of different trading strategies, the extent to which a particular method is able to calibrate the trend following and bias components of each strategy is of particular interest[15]. We therefore begin our discussion by focusing on two free parameter sets, $[g_2, b_2]$ and $[g_2, b_2, g_3, b_3]$, with $g_2, b_2, g_3 \in [0, 1]$ and $b_3 \in [-1, 0]$.

Table 2.5: Calibration Results for Free Parameter Set 2 of the Random Walk Model

	d_1	d_2	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	0.1	0.2	–
GSL-div/PS	0.0311	0.1945	0.0691
GSL-div/KK	0.4572	0.6210	0.5520
MSM/PS	1	1	1.2042
MSM/KK	1	1	1.2042
MIC/PS	0.1684	0.4922	0.3001
MIC/KK	0.1494	0.5264	0.3301
BE	0.1721	0.2751	0.1041

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016).

Table 2.6: Calibration Results for Free Parameter Set 1 of the Brock and Hommes (1998) Model

	g_2	b_2	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	0.6	0.2	–
GSL-div/PS	0.2972	0.0272	0.3486
GSL-div/KK	0.2934	0.0319	0.3497
MSM/PS	0.6550	0.2299	0.0626
MSM/KK	0.6185	0.2331	0.0380
MIC/PS	0.8214	0.2082	0.2216
MIC/KK	0.8575	0.2048	0.2575
BE	0.7243	0.2070	0.1245

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016). All other parameters have been set according to true parameter set 4.1 in Table 2.11.

Overall, the observed trends are similar to those seen when investigating the preceding models, with Bayesian estimation performing reasonably well in both cases and the SMD methods being more inconsistent.

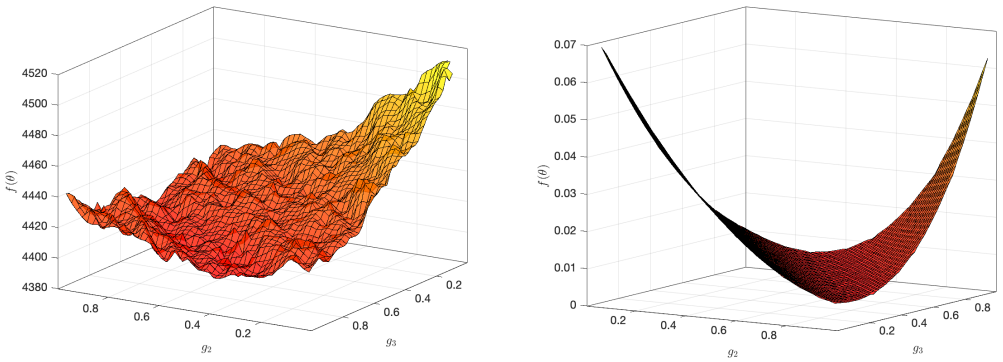
[15]For the majority of the preceding experiments, we provided figures illustrating the behaviours of the posterior distributions and SMD objective functions. For this model, however, we focus primarily on the final estimation results and loss function values for the sake of brevity.

Table 2.7: Calibration Results for Free Parameter Set 2 of the Brock and Hommes (1998) Model

	g_2	b_2	g_3	b_3	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	0.6	0.2	0.7	-0.2	–
GSL-div/PS	0.6047	0	0.5277	0	0.3312
GSL-div/KK	0.5575	0	0.5529	0	0.3216
MSM/PS	1	0.1941	0.5015	-0.2174	0.4469
MSM/KK	0.8742	0.2424	0.2424	-0.2040	0.5352
MIC/PS	0.8105	0.2278	0.7430	-0.2186	0.2173
MIC/KK	0.3663	0.2431	0.8766	-0.2387	0.2985
BE	0.4693	0.2411	0.6404	-0.2330	0.1530

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016). All other parameters have been set according to true parameter set 4.1 in Table 2.11.

Referring to Table 2.6, which presents the results associated with the first free parameter set, we see that while the GSL-div is unable to provide reasonable estimates for g_2 or b_2, the remaining methodologies are all able to determine b_2 to some extent, with the differences in estimation performance largely being due to g_2. A closer inspection of the time series produced by the model suggests that shifts in the value of b_2 have a more significant impact on the overall dynamics than similar shifts in the value of g_2, providing a possible explanation for this behaviour. Interestingly, we see that while Bayesian estimation performs reasonably well, it is in fact MSM that is the best performing method in this case.



(a) MIC (b) MSM

Figure 2.9: SMD objective function cross sections for free parameter set 2 of the Brock and Hommes (1998) model. The values of b_2 and b_3 have been fixed according to MIC/KK and MSM/KK (see Table 2.7) respectively.

Table 2.8: Calibration Results for Free Parameter Set 3 of the Brock and Hommes (1998) Model

	β	$LS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_{true})$
$\boldsymbol{\theta}_{true}$	120	–
GSL-div/PS	104.2303	15.7697
GSL-div/KK	104.3522	15.6478
MSM/PS	117.7987	2.2013
MSM/KK	118.2510	1.7490
MIC/PS	111.3399	8.6601
MIC/KK	112.1062	7.8938
BE	119.5422	0.4578

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016). All other parameters have been set according to true parameter set 4.2 in Table 2.11. β is explored over the range $[0, 150]$.

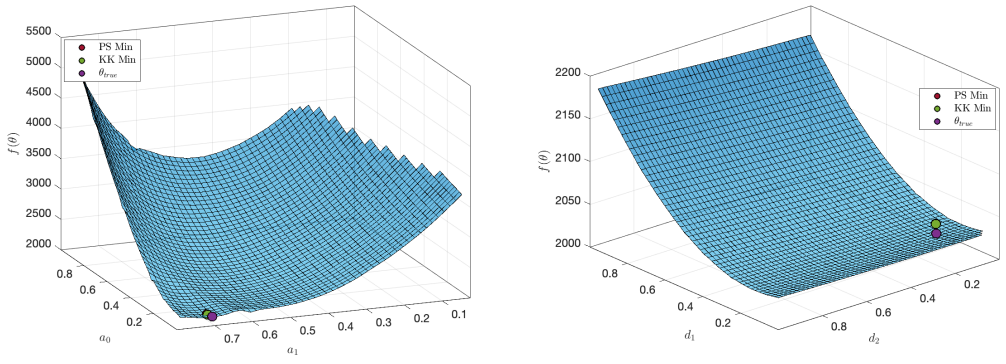
Proceeding to Table 2.7, we observe similar behaviours for the second free parameter set, with b_2 and b_3 generally being more accurately estimated than g_2 or g_3 by most of the considered methodologies, though Bayesian estimation now yields a lower loss function value than MSM. Additionally, as we observed for some of the previously considered models, there are cases where estimates associated with the same objective function but different optimisation algorithms do not agree, specifically the estimates for g_2 and g_3 associated with MSM and the MIC. As before, it can be shown that this largely stems from challenges brought about by the objective functions themselves rather than the failure of the employed optimisation algorithms. Referring to Figure 2.9a, we see that similar arguments apply to those introduced for the second free parameter set of the ARMA$(2, 2)$-ARCH(2) model, with the objective function noise associated with the MIC being the likely culprit. In contrast to this, Figure 2.9b reveals that the MSM objective function surface is in fact smooth. Notice, however, that the global minimum appears to lie within a flat region at the bottom of a u-shaped valley that runs from one end of the parameter space to the other. It would therefore appear that the effect of g_2 and g_3 on the MSM objective function is characterised by a degree of collinearity, making these parameters difficult to estimate using MSM.

Finally, before concluding our comparative experiments, we include one last exercise in which we attempt to calibrate the intensity of choice, β. In contrast to the first and second free parameter sets, where β was set such that the model output exhibited realistic dynamics (see true parameter set 4.1 in Table 2.11), we now consider more extreme values (see true parameter set 4.2 in Table 2.11). This allows us to assess the extent to which the considered methods are able to distinguish between the dynamics that result from different values of β in a highly volatile regime where trader agents rapidly switch between strategies. As can be seen in Table 2.8, the relative performance of each methodology in this scenario is much the same as before.

2.2.1.5 Supplementary Experiments

Given the promising performance of the Bayesian estimation procedure proposed by Grazzini et al. (2017) relative to the tested SMD methods, it would be worthwhile to provide a

more precise explanation for the observed phenomena. Unfortunately, this is a challenging undertaking, primarily due to the fact that the Bayesian estimation methodology differs from the remaining approaches in several fundamental ways. Unlike the remaining methods, it directly constructs a simulation-based approximation to the likelihood function rather than a distance-based measure, employs the likelihood in a Bayesian as opposed to frequentist framework, and makes use of MCMC sampling rather than heuristic optimisation.



(a) ARMA-ARCH model, free parameter set 1 (b) Random walk model, free parameter set 2

Figure 2.10: SML negative log-likelihood surfaces, with PS referring to particle swarm optimisation and KK referring to the approach of Knysh and Korkolis (2016).

Each of the above factors has the potential to affect the overall quality of the resultant estimates and decoupling their effects is nontrivial. Despite this, we attempt to provide at least some explanation by repeating a subset of the previously conducted experiments, this time directly optimising the likelihood in a frequentist framework, leading to what is often referred to as simulated maximum likelihood (SML).

Figure 2.10, which presents the results of these experiments, provides a number of important insights. Firstly, notice that the resultant objective function surfaces are smooth. This is in contrast to the GSL-div and MIC, where objective function noise proved to be a significant obstacle for the optimisation process. Additionally, observe that the obtained estimates are relatively close to the true parameter values. This is particularly relevant in the case of the second free parameter set of the random walk model, since the competing SMD approaches were generally unable to recover appropriate values for these parameters. It would therefore appear that the likelihood construction employed by the method of Grazzini et al. (2017) is a key component of its success, regardless of whether the likelihood is employed within a frequentist or Bayesian framework and regardless of whether it is optimised directly or indirectly used within an MCMC algorithm.

Nevertheless, a closer examination reveals that other factors do also play some part in the method's success. In more detail, Tables 2.9 and 2.10 reveal interesting similarities and contrasts between Bayesian estimation and SML. Firstly, we see that although the SML estimate and posterior mean are very similar in the case of the ARMA$(2, 2)$-ARCH(2) model, there are clear differences between them when considering the random walk. This can be explained as follows. While it may be tempting to assume that Bayesian estimation with uniform priors is equivalent to maximum likelihood estimation, this is in fact not generally

true. Rather, it turns out that this equivalence holds only for the posterior mode and that other point estimates, such as the posterior mean or median, only coincide with the SML estimate in cases where they additionally coincide with the posterior mode (Murphy 2012). In some of the examples considered in this investigation, such as the 5 parameter case of the ARMA(2, 2)-ARCH(2) model (see Figure 2.4), the posterior mode would appear to be a vastly inferior estimate when compared to the posterior mean[16].

Table 2.9: A Comparison of Various Likelihood-based Estimates for Free Parameter Set 1 of the ARMA(2, 2)-ARCH(2) Model

	a_0	a_1
$\boldsymbol{\theta}_{true}$	0	0.7
SML/PS	0.0540	0.7034
SML/KK	0.0471	0.7041
Posterior Mode	0.0472	0.7044
Posterior Mean	0.0449	0.7039

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016). The posterior mode is estimated using KDE.

This points to a potential advantage of the adoption of a Bayesian paradigm. Rather than obtaining a single point estimate, as would be the case in a frequentist framework, we instead obtain a posterior distribution that allows for the calculation of several distinct point estimates. Common point estimates, such as the mean, median, and mode, can be shown to be minimisers of a number of important loss functions, namely the squared error loss,

$$L(\theta_{true}, \hat{\theta}) = (\theta_{true} - \hat{\theta})^2, \tag{2.18}$$

absolute value loss,

$$L(\theta_{true}, \hat{\theta}) = |\theta_{true} - \hat{\theta}|, \tag{2.19}$$

and $0-1$ loss,

$$L(\theta_{true}, \hat{\theta}) = \mathbb{I}(\theta_{true} \neq \hat{\theta}), \tag{2.20}$$

respectively (Murphy 2012).

Our aim when calibrating a given model, at least in the context of this investigation, is the determination of parameter estimates that are as close as possible (in terms of Euclidean distance) to the true parameter values. The posterior mean is thus a natural choice in such a context (given its minimisation of the squared error loss) and this seems to be reflected in the reasonable quality of estimates obtained in all of the considered examples[17]. In contrast to this, the $0-1$ loss function, which is minimised by the posterior mode, makes no distinction between estimates that are close to or far from the true parameter values if none of the considered estimates are exactly equal to the true parameter values. For this reason, the posterior mode would be a less appropriate choice in our context, as illustrated in the 5

[16] Conversely, the posterior mode appears to be a preferable estimate for the second free parameter set of the random walk model, though the difference in performance appears to be quite small.

[17] Of course, the posterior median would be more appropriate if we had instead considered a comparison based on the Manhattan distance. Nevertheless, the mean and median are generally quite similar, with the mode being the point estimate most likely to result in noticeably different parameter values in some cases.

parameter case of the ARMA(2, 2)-ARCH(2) model mentioned above. It therefore seems that, at least in some cases, the posterior mean is a superior estimate when compared to the maximum likelihood estimate (posterior mode).

Table 2.10: A Comparison of Various Likelihood-based Estimates for Free Parameter Set 2 of the Random Walk Model

	d_1	d_2
$\boldsymbol{\theta}_{true}$	0.1	0.2
SML/PS	0.1789	0.1356
SML/KK	0.1763	0.1381
Posterior Mode	0.1712	0.1332
Posterior Mean	0.1721	0.2751

In the above table, PS refers to particle swarm optimisation and KK refers to the approach of Knysh and Korkolis (2016). The posterior mode is estimated using KDE.

Ultimately, a detailed study of the above phenomena is beyond the scope of this investigation and it would be presumptive to imply that any of the above behaviours hold in general. Regardless, there seems to be at least some evidence that while the likelihood construction is a significant contributor to the method's success, its Bayesian nature, which allows for the calculation of various point estimates rather than the single point estimate associated with a frequentist framework, cannot be ruled out.

As a final remark, it is worth noting that when compared to the MIC surfaces in Figures 2.2 and 2.8, the negative log-likelihood in Figure 2.10 is remarkably similar, despite differences in smoothness. This is not entirely surprising, however, as the MIC, by being an estimator of the cross entropy, is directly related to the likelihood. Therefore, as previously suggested, it seems that the MIC, rather than being an inaccurate measure of distance, has an overall shape very similar to that of the likelihood. Unfortunately, in situations where the likelihood is quite flat, the effects of noise inevitably result in reduced calibration performance.

2.2.1.6 Overall Assessment

From the preceding computational experiments, it seems apparent that the Bayesian estimation procedure introduced by Grazzini et al. (2017) consistently outperforms the tested SMD methods. Indeed, we find that the approach produces estimates that are comparable with the true parameter values for every model and free parameter set tested and is associated with loss function values lower than those of competing methods in all but a minority of cases[18].

SMD methods, on the other hand, seem less consistent. While there are indeed a number of cases where each objective function performs well, there are also multiple cases where their performance is greatly compromised, which seems to stem either from inaccuracies in the

[18]While not a major component of our investigation, we provide a rough overview of the computational demands associated with each methodology in Appendix 2.C, which suggests that Bayesian estimation is also competitive in this regard.

dissimilarity measurements or objective function noise that makes it difficult or even impossible to effectively optimise them. Elaborating on this point, the MIC, while demonstrating strong agreement with the likelihood function employed in the Bayesian estimation experiments (see Section 2.2.1.5), is significantly affected by noise. On the other hand, MSM, while yielding smooth objective function surfaces, is limited by its consideration of a set number of summary statistics rather than the original time series data, which, as shown in Figure 2.9b, limits its ability to distinguish between certain parameter combinations. Finally, the GSL-div, which focuses primarily on the relative movements of time series observations and is agnostic to the magnitude of those movements, seems to suffer from both weaknesses.

At this point, it should be noted that this does not call for an outright dismissal of the GSL-div or MIC. As previously stated, both methodologies were originally introduced as model selection tools, with applications to calibration being proposed as extensions, and this investigation has no bearing on their effectiveness in the model selection context. In particular, such an application would typically not require any form of search to be conducted and the noise encountered here is therefore less likely to be of concern.

2.2.2 INET Oxford Housing Market Model

Given that the Bayesian estimation procedure proposed by Grazzini et al. (2017) appears to work well for a wide variety of simple models, it would be natural to ask if similar success can be achieved when attempting to calibrate a more realistic ABM. Along these lines, we now attempt to calibrate a large-scale model of the UK housing market and consider two free parameter sets of size 4 and 19 respectively. The rationale behind these choices is relatively straightforward. By considering only 4 free parameters, we are able to determine whether a degradation in performance is observed for calibration problems of a similar size but with the underlying model being significantly more computationally-expensive and producing more nuanced dynamics. Thereafter, we can additionally test the effect of increasing the cardinality of the parameter space by considering 19 free parameters.

2.2.2.1 Free Parameter Set 1

While there are potentially many parameters that could be considered when constructing our first free parameter set, it is often useful to prioritise parameters that may potentially be strong drivers of the overall model dynamics or are of particular interest to the modeller. In line with this, we have selected *Market Average Price Decay*, *Sale Epsilon*, *P Investor* and *Min Investor Percentile* as the parameters to be calibrated in our first attempt, since each is an important driver of one of the model's key mechanisms.

Unlike previous examples, the model produces panel data as opposed to a single time series. While we would ideally wish to consider the full set of model outputs, computational limitations[19] necessitate that we select a subset for use in our calibration experiments. In the case of the first free parameter set, we have selected a single output, the sale HPI. We justify this choice as follows. Firstly, in comparison to the remaining output variables, the sale HPI is generally the most informative regarding the overall state of the housing market at a particular point in time. Secondly, as we will soon show, even a single output is sufficient to achieve reasonably good estimates for the considered free parameters.

[19] A greater number of outputs would require the generation of additional model Monte Carlo replications in order to ensure that a sufficient number of samples is generated to learn the required density function using KDE.

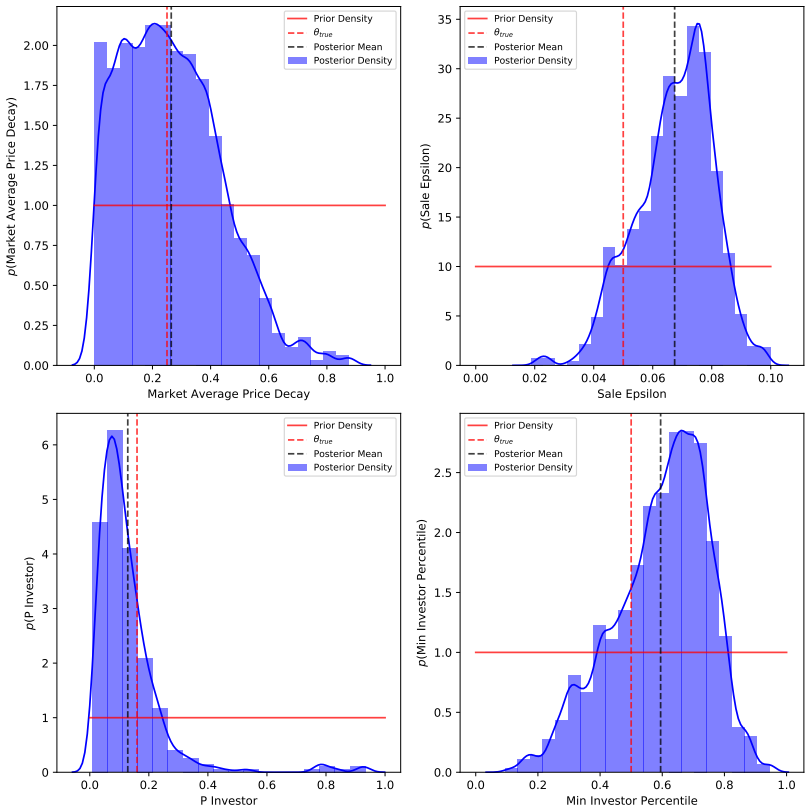


Figure 2.11: Marginal posterior distributions for free parameter set 1 of the housing market model.

Referring to Figure 2.11, which presents the obtained marginal posterior distributions, we see that despite a significant increase in the sophistication of the candidate model, the performance of the method of Grazzini et al. (2017) is still relatively good, with the observed behaviours being more or less consistent with those seen in the 5 and 4 parameter cases of the ARMA$(2, 2)$-ARCH(2) and Brock and Hommes (1998) models respectively (see Tables 2.3 and 2.7).

This is a very promising result and suggests that the method is relatively robust and can be trusted in a wide variety of contexts to reliably calibrate several free parameters. Nevertheless, we will ultimately need to consider a much larger free parameter space, which we do shortly, in order to assess the extent to which this performance is maintained as the number of dimensions is increased.

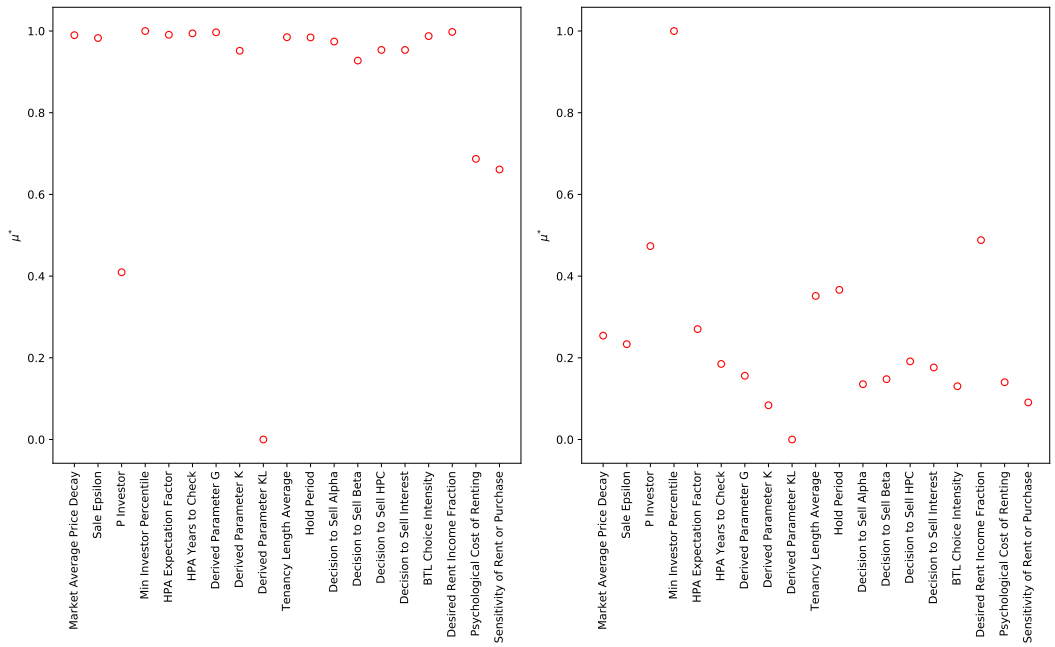
2.2.2.2 Free Parameter Set 2

In Section 1.1.1, we indicated that it often arises that some of the parameters of a given ABM may be amenable to calibration using direct observation. While many of the housing market model's parameters can be set in this way, there are notable exceptions, which form the basis of our second free parameter set. At first glance, it may be tempting to attempt to apply a similar strategy to that employed in our previous experiments, but the dimensionality

of this problem necessitates a more nuanced approach. In more detail, we begin with a sensitivity analysis to ascertain whether the effect of the considered free parameters on the model output is sufficient such that they could be realistically identified by the calibration procedure.

To achieve this, we make use of an extended version of the well-known method of Morris (1991), which is discussed in detail by Campolongo et al. (2007). In brief, the approach produces a measure, $\mu^*(\theta_j)$, for each parameter, $\theta_j, j = 1, 2, \ldots, 19$, that quantifies the sensitivity of a function of interest, f, to changes in θ_j. Here we consider two separate cases for f, one where f is the posterior density function produced by the method of Grazzini et al. (2017), and a second where it is the model output itself. Full details are provided in Appendix 2.A. In contrast to our procedure for the first free parameter set, we now consider two model outputs, the sale HPI (as before) and the rental HPI, which together capture two contrasting, but equally-important aspects of the simulation, the sale and rental markets.

Referring to Figure 2.12a, which presents the results of our sensitivity analysis for the model output, we observe that the majority of the considered free parameters exhibit significant effects on the generated paths. In more detail, 15 of the 19 considered parameters are associated with effects at least 90% as strong as those associated with the most influential parameter, with 3 of the remaining 4 parameters demonstrating weaker but noticeable effects ranging from approximately 40% to 70%. In contrast to this, only one parameter, *Derived Parameter KL*, seems to have a negligible effect on the model output.



(a) Model Output

(b) Posterior Density

Figure 2.12: Sensitivity analysis for free parameter set 2 of the housing market model. The μ^* values in each figure have been divided by the largest value obtained in each experiment, 2178.1669 and 2946.5422 respectively, to provide a relative scale.

Interestingly, Figure 2.12b reveals that the same cannot be said of the posterior density produced by the method of Grazzini et al. (2017). Indeed, we now see a very different spread of μ^* values, with the majority of the considered parameters having noticeably weaker effects when compared to the most influential parameter and the effects of 11 of the considered parameters now being less than 20% as strong. This suggests that the effects of a sizeable number of the parameters on the likelihood function are weaker than their corresponding effects on the original model output. This is not entirely surprising, however. Recall from Section 2.1.3.3 that the method of Grazzini et al. (2017) assumes that the observations of a given time series or set of panel data are independent draws from some distribution and would therefore assign an identical probability to datasets that have the same unconditional distribution. Unfortunately, while this assumption is convenient, it is unlikely to hold in practice and in many cases may result in important trends in the data being ignored. This seems to be reflected here.

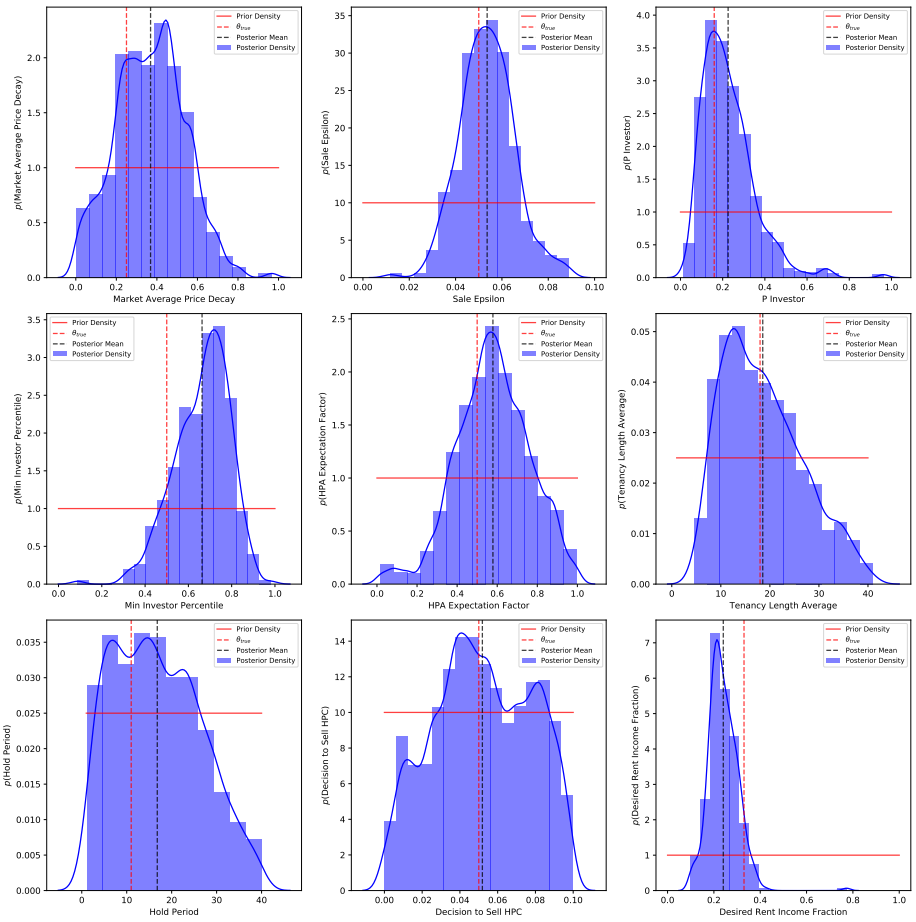


Figure 2.13: Marginal posterior distributions for free parameter set 2 of the housing market model.

Initial attempts to calibrate the full set of 19 free parameters proved to be unsuccessful,

with roughly half of the parameters being unidentifiable. It therefore seems that the effects of at least some of the parameters on the posterior density are indeed not sufficient for the parameters to be identified using the employed methods. We therefore instead attempt to calibrate a total of 9 free parameters, chosen according to the μ^* values (in descending order) presented in Figure 2.12b.

With reference to Figure 2.13, which illustrates the marginal posterior distributions for these parameters, we observe that the overall performance of the method, while not as good as in previous cases, is still reasonable. To be more precise, the posterior means for most of the considered free parameters are generally located in the same area of the parameter space ($10 - 15\%$ of the explored parameter range) as their true values, with the estimates for certain parameters, such as *Sale Epsilon*, *P Investor*, *HPA Expectation Factor*, *Tenancy Length Average*, and *Decision to Sell HPC* being particularly close.

Overall, the methodology appears to be an appropriate choice when tackling more complex models, despite some limitations arising from its independence assumptions, and it is reasonable to expect that the quality of these estimates would improve with the consideration of additional model outputs and longer simulated time series.

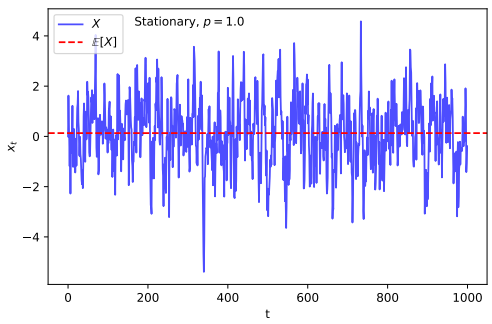
Conclusion

From the preceding discussions, it should be apparent that despite their strong popularity in the existing literature, the overall performance of SMD methods is mixed. While there is indeed evidence that such methods may be effective for simple calibration problems consisting of 1 or 2 free parameters, their reliability rapidly declines as the number of free parameters is increased. This can primarily be attributed to one of two key weaknesses: severe objective function noise that renders convergence to a global minimum unlikely or impossible, even when sophisticated heuristics are employed, or objective functions that are characterised by multiple or an infinite number of global minima, which points to an inability to distinguish between the true parameter values and a large number of alternatives.

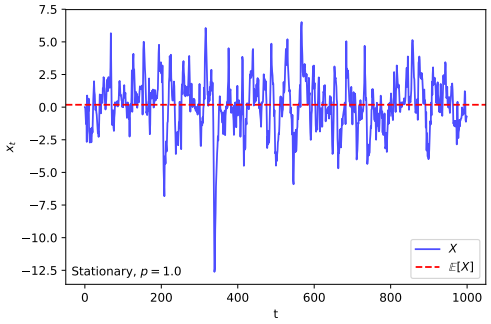
On the other hand, the Bayesian estimation procedure proposed by Grazzini et al. (2017) appears to be far more compelling, showing a degradation in performance only when confronted with a high-dimensional problem involving a large-scale model. Identifying the precise mechanisms that lead to the approach's success is nontrivial, but there appears to be at least some evidence that its superior performance is due to a combination of factors: its well-behaved likelihood approximation and adoption of a Bayesian paradigm. Further investigation would likely be necessary to definitively determine the relative importance of each of the aforementioned factors and whether they are general or specific to the method of Grazzini et al. (2017). Unfortunately, this is well beyond the scope of this thesis and therefore left to future research.

Overall, the relative success of the estimation method introduced by Grazzini et al. (2017) in the conducted experiments makes a compelling case for an increased emphasis on likelihood-based methods, particularly within a Bayesian setting, and we therefore suggest that their development should become a key area of focus in future research. As a first step, the approach introduced by Grazzini et al. (2017) could certainly be updated by allowing for temporal dependencies to be taken into account and this may, in turn, significantly improve the quality of results obtained for larger models.

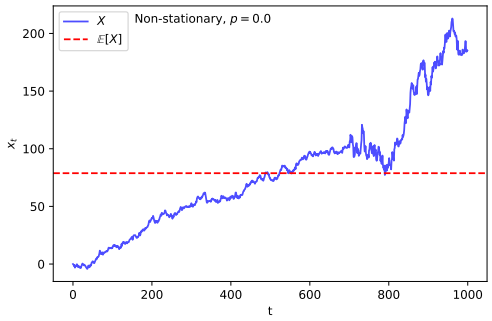
2.A Additional Experimental Details



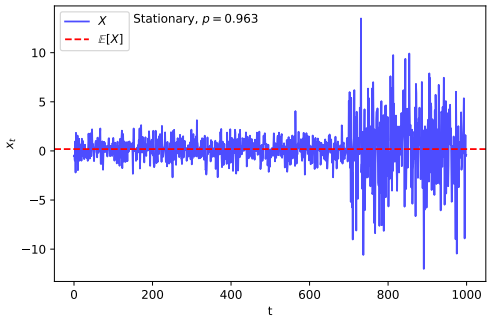
(a) AR(1)



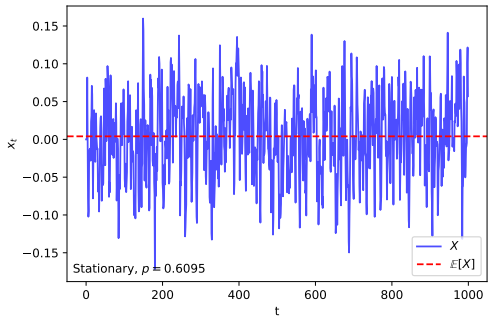
(b) ARMA(2, 2)-ARCH(2)



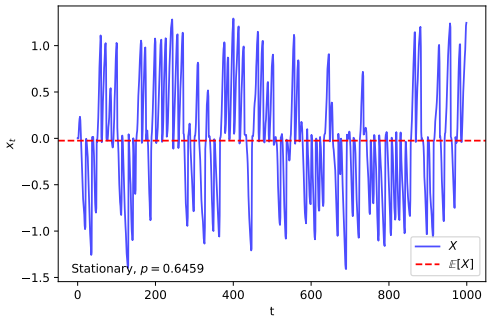
(c) Random walk



(d) Random walk (differences)



(e) Brock and Hommes (1998) (true parameter set 1)



(f) Brock and Hommes (1998) (true parameter set 2)

Figure 2.14: An illustration of the pseudo-true data considered in the calibration experiments. The provided p values are obtained from the stationarity test proposed by Grazzini (2012).

The primary purpose of this appendix is to provide readers with a more detailed specification of each computational experiment than is provided in the main text, with a focus on the generation of the pseudo-true data used for calibration, as well as the hyperparameters chosen for each calibration approach. It is hoped that this will, in principle, aid others in attempting to conduct similar experiments.

2.A.1 Simple Time Series Models

As in the main text, our experiments can be broadly classified as belonging to one of two main classes: attempts at calibrating simple models and attempts at calibrating a large-scale ABM of the UK housing market. We begin with a discussion of the former.

2.A.1.1 Dataset Construction

Recall that we do not make use of empirically-observed data. Rather, we consider data generated by a particular model for a chosen set of parameter values and assess the extent to which the true parameter set can be recovered through the calibration of the model to the artificial data. We therefore begin our series of numerical experiments by generating a pseudo-true dataset for each of the models described in Section 2.1.2, which have been initialised using the parameter values presented in Table 2.11.

Table 2.11: True Parameter Values for the Set of Simple Time Series Models

Model	Parameter Symbols	Selected Values
1	a_1	0.7
2	$a_0, a_1, a_2, b_1, b_2, c_0, c_1, c_2$	$0, 0.7, 0.1, 0.2, 0.2, 0.25, 0.5, 0.3$
3	$\tau, \sigma_1, \sigma_2, d_1, d_2$	$700, 1, 4, 0.1, 0.2$
4.1	$g_1, b_1, g_2, b_2, g_3, b_3, g_4, b_4, r, \beta, \sigma, p^*$	$0, 0, 0.6, 0.2, 0.7, -0.2, 1.01, 0, 0.01, 10, 0.04, 0$
4.2	$g_1, b_1, g_2, b_2, g_3, b_3, g_4, b_4, r, \beta, \sigma, p^*$	$0, 0, 0.9, 0.2, 0.9, -0.2, 1.01, 0, 0.01, 120, 0.04, 0$

The model numbers above correspond to the order in which the models were introduced in Section 2.1.2. In contrast to the remaining models, we consider two cases for model 4, the Brock and Hommes (1998) model, the first of which represents a more realistic intensity of choice and the second of which represents a more extreme regime.

It should be noted that each calibration method makes certain assumptions regarding the data and we should therefore ensure that all such assumptions are satisfied by each artificial dataset before proceeding any further. The most notable of these assumptions, and one common to the majority of the considered methods, is that of stationarity. Referring to Figure 2.14, where we present the artificial data obtained for the selected parameter values, we see that the assumption of stationarity is indeed satisfied in almost all cases, with the only exception being the random walk. This is easily addressed, however, since the output of the random walk can be transformed to induce stationarity by considering the series of first differences[20].

[20]This transformation is applied to all series generated by the random walk model in all calibration experiments.

2.A.1.2 Experimental Procedure

While Section 2.1 provides a relatively complete high-level description of the overall experimental procedure, it is still necessary to clarify a number of finer details.

Firstly, we are required to choose appropriate lengths for both the empirically-observed and model-simulated time series. Figure 2.14 reveals that we have selected an empirical time series length of $T_{emp} = 1000$ for all models, which we believe strikes a good balance between being sufficiently long for robust calibration exercises, while still being sufficiently short such that time series data of a similar length could be obtained empirically. We have similarly selected a simulated time series length of $T_{sim} = 1000$, since some of the considered methodologies, namely the GSL-div, implicitly assume that $T_{emp} = T_{sim}$.

Secondly, as has been pointed out by Grazzini et al. (2017), the cost of generating simulated data is generally the most significant bottleneck for realistic calibration exercises. Therefore, in order to ensure the overall fairness of our comparative experiments, we allocate each method an identical fixed budget of 2500000 model calls (simulated time series realisations) for the purposes of producing a final point estimate. The budget is utilised as follows:

- Each SMD objective function involves the comparison of an ensemble of $R = 250$ model Monte Carlo replications[21] with the (pseudo-true) empirical data and is evaluated a total of 1000 times by the employed optimisation algorithm when being minimised in a given optimisation run[22]. This corresponds to $N_{particles} = 40$ and $T_{generations} = 25$ for particle swarm and $N_{LHS} = 750$ and $N_{CORS} = 250$ for the approach of Knysh and Korkolis (2016). The optimisation process is repeated for a total of 10 runs and the best value of $\boldsymbol{\theta}$ found across the optimisation runs is returned.

- Each evaluation of the posterior density involves the generation of an ensemble of $R = 100$ model Monte Carlo replications and a total of $S = 5000$ such evaluations are performed when applying the adaptive Metropolis-Hastings algorithm proposed by Griffin and Walker (2013) to generate a single chain of S sample sets[23]. We generate 5 such chains and combine them to form a single, larger collection of 17500 sample sets[24] that describes the posterior distribution.

From the above, it should be apparent that each SMD objective function evaluation generally requires a greater number of Monte Carlo replications than a single evaluation of the posterior density in the Bayesian estimation framework proposed by Grazzini et al. (2017). This is because most SMD methods involve the estimation of quantities such as moments or occurrence frequencies for each series, which are then averaged over the ensemble. Each additional series therefore provides only one additional value of the quantity of interest. In the case of Bayesian estimation, however, each new series adds an additional T_{sim} observations to the sample to which KDE is applied.

[21] For series of length 1000, Barde (2017) suggests that ensemble sizes of 250 or 500 be considered when employing the MIC, motivating our choice. In general, the ensemble size needs to strike a balance between being sufficiently large to compensate for Monte Carlo variance, while still being sufficiently small such that similar exercises could be conducted in the context of large-scale, computationally-expensive models. Since most institutional clusters are likely to have in the order of 100 to 200 cores, we believe this to be a reasonable choice.

[22] This was sufficient for convergence in all cases.

[23] Each sample set consists of $N = 70$ samples.

[24] In all cases, we observed convergence after only a few hundred iterations and thus discarded the first 1500 sample sets as part of a burning-in period.

2.A.1.3 Method Hyperparameters

It often arises that calibration methods have parameters of their own that need to be carefully chosen in order to maximise performance. In our investigation, this applies to two of the considered methodologies, the GSL-div and MIC.

Table 2.12: MIC Hyperparameter Values and Validity Tests

Model	b_l	b_u	r	L	Uniform (K–S Test)	Uncorrelated (Pearson)
1	-5	5	5	3	Yes, $p = 0.5073$	Yes, $p = 0.0869$
2	-30	30	7	2	Yes, $p = 0.6858$	Yes, $p = 0.0989$
3	-15	15	6	3	Yes, $p = 0.2983$	Yes, $p = 0.1423$
4.1	-1	1	8	2	Yes, $p = 0.6213$	Yes, $p = 0.7634$
4.2	-1.5	1.5	8	2	Yes, $p = 0.0550$	Yes, $p = 0.9198$

The model numbers above correspond to the order in which the models were introduced in Section 2.1.2. In contrast to the remaining models, we consider two cases for model 4, the Brock and Hommes (1998) model, the first of which represents a more realistic intensity of choice and the second of which represents a more extreme regime.

In the case of the GSL-div, this choice is relatively straightforward, with Lamperti (2018a) providing evidence that $b = 5$ and $L = 6$ are appropriate choices for most problems. The setting of the MIC's hyperparameters is, however, slightly more nuanced, since suitable parameter settings may vary from model to model. Fortunately, Barde (2017) introduces a set of properties that can be tested to ensure that the chosen values are appropriate. Our selected values and the results of related tests are presented in Table 2.12.

It should be noted that the choice of $L = 2$ for two of the above models is necessitated due to memory limitations. Such a setting is nonetheless reasonable, however, since Barde (2017) demonstrates the relative robustness of the MIC to the choice of L.

2.A.2 INET Oxford Housing Market Model

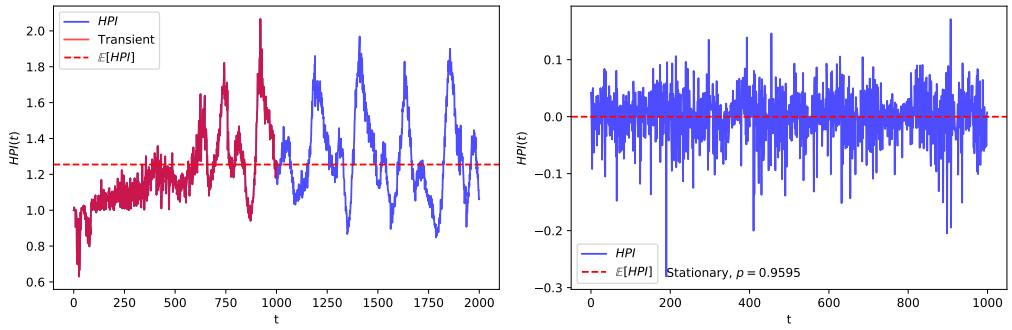
Having now described our initial comparative experiments in more detail, we proceed with a discussion of the experiments involving a large-scale model of the UK housing market.

2.A.2.1 Dataset Construction

Unlike the models we have previously considered, the housing market model produces panel data as opposed to a single time series. As discussed in more detail in the main text, we select a subset of these outputs for use in our calibration experiments, specifically the sale HPI and rental HPI, which represent the aggregate behaviours of two very different aspects of the associated housing market, the sale and rental sectors respectively. Our focus here is on the generation of our pseudo-true dataset, which has been generated using the model's default parameter values, which are listed at `github.com/adrian-carro/spatial-housing-model/blob/last-before-spatial/src/main/resources/config.properties`.

Referring to Figure 2.15a, we see that the consideration of sale HPI results in a number of complications. Firstly, notice that the sale HPI time series is characterised by an initial

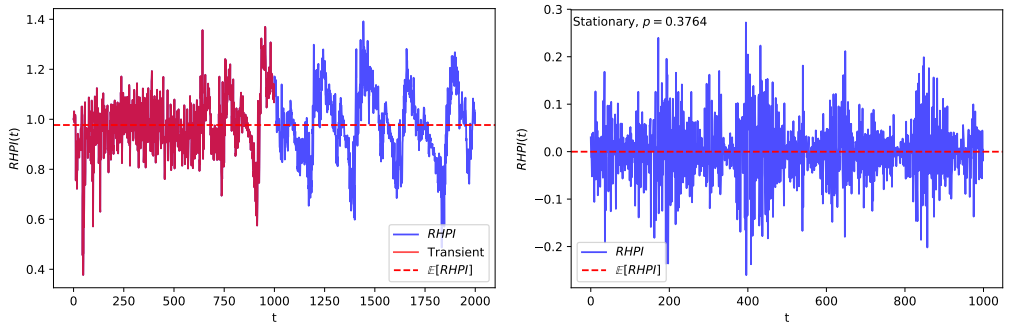
transient period, followed by cyclical dynamics that persist for the remainder of the simulation. Since this transient period corresponds to the model's initialisation procedure rather than its true dynamics, it should be discarded. Secondly, we see that the model output is non-stationary, hence we consider the time series of first differences, as in the case of the random walk model (see Figure 2.15b). Analogous issues also apply to the rental HPI time series (see Figures 2.16a and 2.16b), leading us to apply an identical transformation.



(a) Original Time Series

(b) First Difference Series with the Transient Period Discarded

Figure 2.15: Sale HPI time series for the model's default parameter values. The provided p values are obtained from the stationarity test proposed by Grazzini (2012).



(a) Original Time Series

(b) First Difference Series with the Transient Period Discarded

Figure 2.16: Rental HPI time series for the model's default parameter values. The provided p values are obtained from the stationarity test proposed by Grazzini (2012).

2.A.2.2 Experimental Procedure

Our experimental procedure remains largely unchanged from that employed during the calibration of the set of simple time series models. Nevertheless, the increased computational cost and cardinality of the parameter space associated with the housing market model requires us to make a number of minor adjustments. Firstly, we reduce the size of the ensemble

of Monte Carlo replications to $R = 50$ for the experiments associated with both free parameter sets, such that the amount of time taken to perform a single evaluation of the posterior density function is more computationally-tractable. Secondly, we increase the number of sample sets generated by the adaptive Metropolis-Hastings algorithm to 15000 per chain for the calibration experiment associated with the second free parameter set[25], since a greater number of iterations is required before convergent behaviour can be observed. In line with this, we also increase the burning-in period to 7500 iterations.

2.A.2.3 Sensitivity Analysis

When attempting to calibrate 19 of the housing market model's parameters, we begin with an analysis of the sensitivity of both the posterior density and the model output to changes in the parameter values. This is achieved through the application of a simple, yet well-known sensitivity analysis procedure originally introduced by Morris (1991), which proceeds according to the following steps:

1. The explored range of each parameter is discretised to form a set of 100 equispaced points, resulting in a 100×19 grid that represents the space of feasible parameter values.

2. 750 of these grid points are sampled at random, with each point, $\boldsymbol{\theta}^i = [\theta_1^i, \theta_2^i, \ldots, \theta_{19}^i], i = 1, 2, \ldots, 750$, being the start of what is referred to as a Morris (1991) trajectory.

3. For each starting point, $\boldsymbol{\theta}^i$, the remainder of the associated trajectory is iteratively constructed:

   (a) Leaving all other parameter values fixed, a single parameter selected at random, θ_j^i, is shifted by a fixed number of grid points[26], Δ_j, to yield the next point in the trajectory, $[\theta_1^i, \theta_2^i, \ldots, \theta_j^i + \Delta_j, \ldots, \theta_{19}^i]$.

   (b) This process is then repeated, once for each of the 19 free parameters and with the shift being applied to the most recently generated point in the trajectory during each repetition, to yield a complete trajectory of 20 points, $[\theta_1^i, \theta_2^i, \ldots, \theta_{19}^i], [\theta_1^i, \theta_2^i, \ldots, \theta_j^i + \Delta_j, \ldots, \theta_{19}^i], \ldots, [\theta_1^i + \Delta_1, \theta_2^i + \Delta_2, \ldots, \theta_{19}^i + \Delta_{19}]$.

4. Once the trajectories have been constructed, some function of interest, f, which we discuss in more detail shortly, is evaluated at all $750 \times 20 = 15000$ points.

5. Using the values of f calculated during the previous step, it is then possible to generate, for each parameter θ_j, 750 realisations of the quantity $d(\theta_j)$, which is of the form

$$d(\theta_j) = \frac{f(\theta_1, \theta_2, \ldots, \theta_j + \Delta_j, \ldots, \theta_{19}) - f(\theta_1, \theta_2, \ldots, \theta_{19})}{0.1}. \tag{2.21}$$

6. In the standard version of the method, we would then proceed by calculating the mean and variance of $d(\theta_j)$ over the 750 realisations, yielding measures of the strengths of individual parameter and parameter interaction effects respectively. Following the

[25] For the experiments associated with the first free parameter set, our settings remain unchanged from those employed in the comparative experiments involving simple models.

[26] In practice this is a shift of 10 grid points, equivalent to 10% of the explored parameter range.

suggestion of Campolongo et al. (2007), however, we consider a more robust measure that incorporates both types of effect simultaneously, the mean of $|d(\theta_j)|$, which we call $\mu^*(\theta_j)$.

Finally, as suggested above, we consider two cases for f:

1. The log-posterior density, generated according to the approach of Grazzini et al. (2017) using an ensemble of $R = 50$ model Monte Carlo replications.

2. The model output. In this case, we generate a single model realisation[27], $\boldsymbol{X}(\theta_1, \theta_2, \ldots, \theta_{19})$, for each of the 15000 points considered in the sensitivity analysis, with the same random seed used throughout. We then set the numerator in Eqn. 2.21 to be $||\boldsymbol{X}(\theta_1, \theta_2, \ldots, \theta_j + \Delta_j, \ldots, \theta_{19}) - \boldsymbol{X}(\theta_1, \theta_2, \ldots, \theta_{19})||_1$, where $||\boldsymbol{A}||_1$ is the entry-wise L1 norm of matrix $\boldsymbol{A}$.

2.B Sampling Algorithm Modifications

It should be noted that while our employed adaptive Metropolis-Hastings algorithm is almost identical to that introduced by Griffin and Walker (2013), some small, but important modifications have been made.

To elaborate, although the original routine delivers robust convergence to the posterior without careful manual step-size adjustments and is able to handle complex multimodal distributions, the unmodified procedure can introduce a degree of noise into the resultant samples. In more detail, notice from the discussion in Section 2.1.3.4 that the N points in the initial sample set are drawn uniformly from the space of feasible parameter values and, with each subsequent step of the algorithm, are replaced, one-by-one, to generate subsequent sample sets. After a sufficient number of these iterations, the points in the current sample set should be distributed according to the posterior distribution. In some cases, however, and particularly when the posterior is very narrow relative to the prior, a very small number of low-likelihood points (usually only a single point) at a significant distance from the actual posterior samples may remain in the current sample set, regardless of the number of algorithm iterations considered.

This occurs because a new proposal is learnt using the points in the current sample set during each iteration of the procedure. To illustrate, consider the case where all but one of the points in the current sample set have converged to be distributed according to the posterior distribution and the single remaining point is relatively far away from these posterior samples. Calling this point $\boldsymbol{\theta}_s^{(n)}$, notice that $\tilde{p}\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(n-1)}, \boldsymbol{z}, \boldsymbol{\theta}_s^{(n+1)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)$ in the numerator of Eqn. 2.16 would be extremely small, or practically 0 given finite machine precision, as $\boldsymbol{\theta}_s^{(n)}$ could not be reached using a proposal based on the remaining points. Therefore, even if $p\left(\boldsymbol{z}\middle|\boldsymbol{X}\right)/p\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{X}\right) >> 1$, or, in other words, $\boldsymbol{\theta}_s^{(n)}$ were a low-likelihood point far away from the true posterior, the acceptance probability would always be 0 and $\boldsymbol{\theta}_s^{(n)}$ would not be replaced by any subsequent candidate points, even if $S \to \infty$.

In practice, such low-likelihood points tend to represent a very small fraction of the final samples (typically significantly less than 1%) and in practically all cases have a very small

[27] Each realisation is a 1000×2 matrix with columns corresponding to the sale HPI and rental HPI respectively.

effect on the accuracy of point estimates, such as the posterior mean. For this reason, the original algorithm is more than adequate for the purposes for which it would be employed in this chapter[28]. Nevertheless, we did find that these points do, in some cases, have a noticeable effect on the accuracy of estimates of the posterior variance. This could be considered problematic in later chapters, specifically Chapters 3 and 4, where the posterior standard deviation becomes a relevant concern, both as a measure of estimation uncertainty and as an input to various model selection protocols. We therefore introduce a simple, but principled correction to the algorithm to account for the possibility of encountering invalid samples of the type described above.

In brief, we observe that since all points in the current sample set should eventually converge to be distributed according to the posterior, any point $\boldsymbol{\theta}_s^{(n)}$ in the current sample set should have a non-negligible probability of being drawn from the distribution implied by the points in the current sample set, even if $\boldsymbol{\theta}_s^{(n)}$ were to be removed and replaced by another candidate. In other words, $\tilde{p}\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(n-1)}, \boldsymbol{z}, \boldsymbol{\theta}_s^{(n+1)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)$ should not be dramatically smaller than $\tilde{p}\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)$. If the opposite were instead true and the former was indeed several orders of magnitude smaller, this would imply that the probability of drawing $\boldsymbol{\theta}_s^{(n)}$ dramatically decreases if it is not contained in the sample set and that it is therefore unlikely to have been sampled from the same distribution as the remaining points.

In line with the above, and before determining the acceptance probability of a candidate point as in Eqn. 2.16, we first determine if

$$\frac{\tilde{p}\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(n-1)}, \boldsymbol{z}, \boldsymbol{\theta}_s^{(n+1)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)}{\tilde{p}\left(\boldsymbol{\theta}_s^{(n)}|\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N)}\right)} < \epsilon, \tag{2.22}$$

where $\epsilon > 0$ and $\epsilon << 1$.

If the above condition holds, this would imply that $\boldsymbol{\theta}_s^{(n)}$ is likely one of the aforementioned problematic points and should thus be replaced, leading us to set $\alpha = 1$. If, however, the above condition does not hold, we simply calculate α as in Eqn. 2.16.

Naturally, the above requires us to choose an appropriate value for ϵ. While any relatively small value could be chosen, we found 10^{-8} to be an appropriate choice for all of the considered cases[29].

As a final remark, Figures 2.17a and 2.17b provide a simple comparison of the unmodified and modified algorithms respectively in the context of a normal distribution, $\mathcal{N}(-2, 4)$. Here, we see that while both sampling algorithms perform well and produce very similar mean estimates, the unmodified algorithm is characterised by the presence of a very small number of the aforementioned low-likelihood points (see $x \simeq 13$ and $x \simeq -23$) and produces a noticeably less accurate standard deviation estimate, justifying the use of the augmented version. Additionally, Figure 2.17c applies our version of the algorithm to a simple mixture

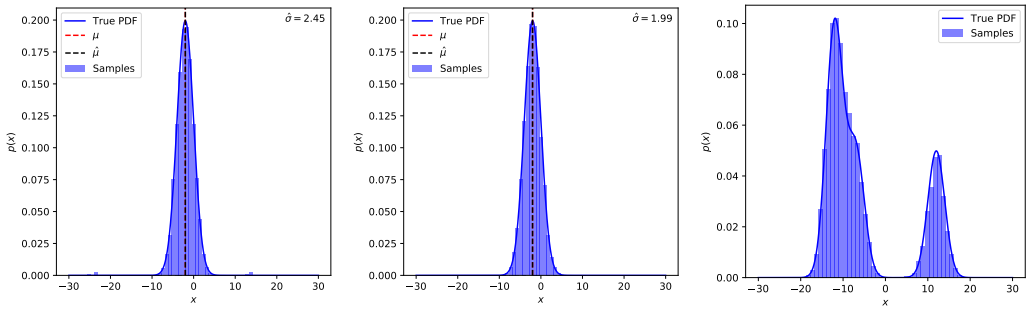
[28] It is in fact the unmodified algorithm that is used in Platt (2020), which also acts as the basis for this chapter. Note, however, that for the purposes of this thesis, we have rerun the majority of the Bayesian estimation experiments using the updated algorithm in order to remain consistent with later chapters. Any resultant changes were found to be extremely negligible, and have no bearing on the qualitative interpretations of the results.

[29] This choice is also extremely conservative, meaning that the adjustment is very rarely applied and only affects points which are extremely far from the remainder of the sample set and only after many initial iterations of the algorithm.

model,

$$\sum_{i=1}^{3} \alpha_i \mathcal{N}(\mu_i, \sigma_i^2), \tag{2.23}$$

where $\mu_1 = -12$, $\mu_2 = -7$, $\mu_3 = 12$, $\sigma_1, \sigma_2, \sigma_3 = 2$, $\alpha_1 = 0.5$, and $\alpha_2, \alpha_3 = 0.25$, and demonstrates that the introduced modifications do not have any impact on the ability of the algorithm to sample multimodal posteriors.



(a) Normal, unmodified (b) Normal, modified (c) Mixture, modified

Figure 2.17: Examples of the application of both the modified and unmodified versions of the adaptive Metropolis-Hastings algorithm proposed by Griffin and Walker (2013) to the problem of sampling known distributions. All examples make use of samples drawn from 5 separate chains, each with $S = 5000$ and a burn-in period of 1500 sample sets. μ is the true mean of the distribution and $\hat{\mu}$ and $\hat{\sigma}$ are the sample mean and standard deviation respectively.

2.C Computational Time Experiments

On the whole, this investigation focuses primarily on assessing calibration methodologies according to the quality of the estimates they produce. While such comparisons are indeed important, one may wonder whether improvements in calibration performance are in fact linked to an increase in computational expense. This section aims to provide some insights into this question.

Ideally, this would be assessed by timing entire calibration experiments, from start to finish, in an attempt to determine the relative length of time required to generate an estimate. Unfortunately, this is not possible here due to non-trivial constraints. In more detail, all of the considered methodologies are associated with significant computational demands and in many cases a single calibration experiment would take several days to run to completion if executed in a serial environment. For this reason, the use of large-scale parallel architectures is required, with a mix of on-site and cloud computing resources being used in our case and both types of resource being subject to queuing and time-varying availability. Therefore, the timing of the actual experiments in their entirety would be largely uninformative, since the amount of computational resources allocated to each algorithm is dynamic and primarily determined by external factors.

We can, however, still provide a rough estimate of the time required for a given calibration approach to generate an estimate for a specific model. We achieve this by employing a simple procedure, capable of being run in a controlled environment, that proceeds as follows:

1. For all of the considered methods, we find that the overheads associated with the optimisation or sampling algorithms are generally negligible compared to the time taken to evaluate the objective function or posterior density. Therefore, the time taken to perform a single objective function or posterior density evaluation is the key driver of the overall computational time required.

2. In line with the above, we generate 100 candidate parameter sets using latin hypercube sampling and evaluate the objective function or posterior density at each of these samples, each time measuring the time taken to perform the evaluation and generate the required data.

3. Each of the obtained measurements is then multiplied by the total number of times the objective function or posterior density is called in a complete experiment, yielding an estimate of the total computational time required had the experiment not been conducted in a parallel environment.

4. We finally report the mean and standard deviation of these approximate total experiment times, which can be compared for different models and calibration methods.

Of course, the above scheme results in a very coarse approximation, but we nonetheless believe it to provide meaningful insights.

Table 2.13: Estimated Total Experiment Time for Various Models and Calibration Methodologies

	AR		ARMA-ARCH		RW		BH	
GSL-div	3261	(289)	3444	(301)	3247	(276)	4988	(279)
MSM	12264	(1043)	11700	(972)	12067	(994)	13884	(1178)
MIC	67633	(6487)	81976	(7205)	97593	(9974)	112540	(3164)
BE	1992	(222)	2622	(312)	2160	(234)	2894	(462)

All values are in minutes with standard deviations in brackets. For each model, the experiment times shown are associated with attempts to calibrate the largest considered free parameter set.

Now, referring to Table 2.13, which presents the approximate experiment times calculated according to this process, we see that the Bayesian estimation procedure proposed by Grazzini et al. (2017) is also competitive in terms of computational time. This is not entirely surprising, however, as the proposed Bayesian estimation procedure is rather simple and involves little more than a single application of KDE during each evaluation of the posterior density. In contrast to this, some of the competing SMD methodologies, such as the GSL-div and particularly the MIC, instead rely on the application of discretisation procedures and a large number of subsequent operations to each series in the ensemble of model Monte Carlo replications, which seems to have resulted in an increase in the associated computational demands relative to Bayesian estimation.

MSM, on the other hand, requires a more nuanced discussion. As it is applied here, we estimate the moment vector for each series in an ensemble of 250 model Monte Carlo replications and take an average over the resultant ensemble of moment vectors when evaluating the objective function for a given parameter set, which unfortunately turns out to

be computationally-expensive. In certain situations, such as in the work of Chen and Lux (2018), where the moments are ergodic when estimated using data generated by the candidate model, it would be possible to consider a single, though slightly longer simulated series rather than an ensemble and achieve a significant computational speedup of several orders of magnitude. While assumptions of ergodicity do hold for a number of simpler models, this is often not the case for those of a larger scale. Since our aim here is not the calibration of simple models, but rather using them as a testing ground for methods that could potentially be applied to large-scale alternatives, we believe that leveraging ergodicity assumptions would render our investigation largely uninformative in this regard.

As a final remark, it should also be noted that the time associated with the actual simulation of the models is negligible, with the operations that the various methodologies apply to the data once it is generated being the primary source of computational expense[30]. In more realistic contexts involving models that must be run for several minutes in order to generate a single output realisation, such as in the calibration of the housing market model considered in this investigation, model simulation time is likely to dominate. Therefore, while the differences in performance showcased here may seem quite significant, they would typically be negligible in a more realistic context. For this reason, a more accurate picture of the relative computational demands of each method is likely to be sketched by the minimum number of Monte Carlo replications, R, required to generate reasonable results, the determination of which is nontrivial and beyond the scope of this investigation.

[30]The variance in the objective function evaluation time from model to model is largely due to the effect of the dynamics present in the data on the time taken to process it. As an example, the MIC is required to draw probabilities from the constructed context trees when calculating the binary log scores used to estimate the cross entropy of the empirical data. Differing levels of complexity in the output dynamics ultimately affect how deep into a given tree this search is likely to go, affecting the associated computational demands.

Chapter 3

Bayesian Estimation of Economic Simulation Models using Neural Networks

In Chapter 1, we indicated that a major shortcoming of the existing ABM estimation literature is the lack of a standard benchmark against which to compare the performance of new methods, which has resulted in most approaches having traditionally only been tested in isolation and comparative exercises having been relatively rare. For this reason, we compared a number of prominent estimation techniques in Chapter 2 and found, rather surprisingly, that the Bayesian estimation procedure proposed by Grazzini et al. (2017) consistently outperformed a number of prominent SMD methods in a series of head-to-head tests, despite its relative simplicity and the high levels of popularity enjoyed by SMD methods. We therefore argued that more interest in Bayesian methods is warranted and suggested that increased emphasis should be placed on their development, with particular reference to the method of Grazzini et al. (2017), which, despite performing well in our considered tests, makes very strong independence assumptions and fails to take important temporal dependencies into account.

In line with the above findings and recommendations, we now introduce a method for the Bayesian estimation of economic simulation models that relaxes a number of the assumptions made by the approach of Grazzini et al. (2017) through the use of a neural network-based likelihood approximation. We then benchmark our proposed methodology through a series of computational experiments and finally conclude with discussions related to practical considerations, such as the setting of the method's hyperparameters and the associated computational costs[1].

3.1 Estimation and Experimental Procedures

In this section, we introduce a number of the essential elements of our investigation, including a brief discussion of the fundamentals of Bayesian estimation, a description of the approach

[1]This chapter is based on the published article: Platt, D., 2021. Bayesian estimation of economic simulation models using neural networks. Comput. Econ. 10.1007/s10614-021-10095-9. Supplementary materials are available at: `https://github.com/D-Platt/alenn`.

of Grazzini et al. (2017) (our chosen benchmark), and an introduction to our proposed estimation methodology. While some of these ideas may have been briefly touched on in Chapter 2, we now provide a more comprehensive discussion as a result of our exclusive focus on Bayesian methods.

3.1.1 Bayesian Estimation of Simulation Models

For our purposes, we consider a simulation model to be any mathematical or algorithmic representation of a real-world system capable of producing time series (panel) data of the form

$$\boldsymbol{X}^{sim}(\boldsymbol{\theta}, T, i) = \left[\boldsymbol{x}_{1,i}^{sim}(\boldsymbol{\theta}), \boldsymbol{x}_{2,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{T,i}^{sim}(\boldsymbol{\theta})\right], \tag{3.1}$$

where $\boldsymbol{\theta}$ is a model parameter set in the space of feasible parameter values, T is the length of the simulation, i represents the seed used to initialise the model's random number generators, and $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta}) \in \mathbb{R}^n$ for all $t = 1, 2, \ldots, T$.

In general, estimation or calibration procedures aim to determine appropriate values for $\boldsymbol{\theta}$ such that $\boldsymbol{X}^{sim}(\boldsymbol{\theta}, T, i)$ produces dynamics that are as close as possible to those observed in an empirically-measured equivalent,

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T], \tag{3.2}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ for all $t = 1, 2, \ldots, T$.

Bayesian estimation attempts to achieve the above by first assuming that the parameter values follow a given distribution, $p(\boldsymbol{\theta})$, which is chosen to reflect one's prior knowledge or beliefs regarding the parameter values. This is then updated in light of empirically-measured data, yielding a modified distribution, $p(\boldsymbol{\theta}|\boldsymbol{X})$, called the posterior. Bayesian estimation can therefore be framed in terms of Bayes' theorem as follows:

$$p(\boldsymbol{\theta}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{X})}. \tag{3.3}$$

Unfortunately, obtaining an analytical expression for the posterior is typically not feasible. Firstly, the normalisation constant, $p(\boldsymbol{X})$, is unknown or determining it is nontrivial. Secondly, the likelihood, $p(\boldsymbol{X}|\boldsymbol{\theta})$, is intractable for most simulation models, particularly large-scale macroeconomic ABMs. Nevertheless, these limitations can be overcome to some extent. As an example, Grazzini et al. (2017) provide a method for approximating $p(\boldsymbol{X}|\boldsymbol{\theta})$ for a particular value of $\boldsymbol{\theta}$, which then allows us to evaluate the right-hand side of

$$p(\boldsymbol{\theta}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{3.4}$$

The above may then be used along with MCMC methods, such as the Metropolis-Hastings algorithm, to sample the posterior. This is possible since most MCMC techniques only require that we are able to determine the value of a function proportional to the density function of interest rather than the density function itself. It should be apparent, however, that the overall estimation error will depend critically on the method used to approximate the likelihood.

3.1.2 The Approach of Grazzini et al. (2017)

As previously discussed in Section 2.1.3.3, Grazzini et al. (2017) provide a method to approximate the likelihood for simulation models, which we now elaborate upon further.

In essence, the approach is based on the assumption that, for all $t \geq \tilde{T}$, we reach a statistical equilibrium such that $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})$ fluctuates around a stationary level, $\mathbb{E}[\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})|t \geq \tilde{T}]$, which allows us to further assume that $\boldsymbol{x}_{\tilde{T},i}^{sim}(\boldsymbol{\theta}), \boldsymbol{x}_{\tilde{T}+1,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{T,i}^{sim}(\boldsymbol{\theta})$ constitutes a random sample from a given distribution[2]. It is then possible to determine a density function that describes this distribution, which we denote by $\tilde{f}(\boldsymbol{x}|\boldsymbol{\theta})$, using KDE, finally allowing us to approximate the likelihood of the empirically-sampled data[3] for a given value of $\boldsymbol{\theta}$ as follows:

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{t=1}^{T} \tilde{f}(\boldsymbol{x}_t|\boldsymbol{\theta}). \tag{3.5}$$

It should be apparent that the above results in a simple strategy that is easy to apply in most contexts. It must be noted, however, that this is largely made possible through strong assumptions that seldom hold in practice. In more detail, notice that ordered time series (panel) data is essentially being treated as an i.i.d. random sample, implying that $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta}) \perp \boldsymbol{x}_{1,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{t-1,i}^{sim}(\boldsymbol{\theta})$ for all $t = 2, 3, \ldots, T$. Unfortunately, such independence assumptions do not hold for most simulation models, since $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})$ is likely to be dependent on at least some of the previously realised values, whether this dependence is explicit or mediated through latent variables. Additionally, such assumptions result in a likelihood function that makes no distinction between $\boldsymbol{\theta}$ values that result in identical unconditional distributions but differing temporal trends. Since many economic simulation models and particularly large-scale macroeconomic ABMs produce datasets that are characterised by seasonality or structural breaks, there is likely to be some impact on the quality of the resultant parameter estimates.

Nevertheless, Chapter 2 demonstrates that despite the above shortcomings, the method of Grazzini et al. (2017) is able to provide reasonable parameter estimates in many contexts, while also outperforming several more sophisticated SMD methods. This warrants further investigation and naturally leads one to ask whether relaxing the required independence assumptions would allow for the construction of a superior Bayesian estimation method.

3.1.3 Likelihood Approximation using Neural Networks

We now begin our discussion of a relatively simple extension to the likelihood approximation procedure proposed by Grazzini et al. (2017) that is capable of capturing some of the dependence of $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})$ on past realised values. As a starting point, we assume that

$$p\left(\boldsymbol{x}_{t,i}^{sim}\middle|\boldsymbol{x}_{1,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim} : \boldsymbol{\theta}\right) = p\left(\boldsymbol{x}_{t,i}^{sim}\middle|\boldsymbol{x}_{t-L,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim} : \boldsymbol{\theta}\right) \tag{3.6}$$

for all $L < t \leq T$, implying that $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})$ depends only on the past L realised values. Our task, therefore, is the estimation of the above conditional densities,

$$\tilde{f}\left(\boldsymbol{x}_{t-L,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim}, \boldsymbol{x}_{t,i}^{sim}, \boldsymbol{\phi}\right) \simeq p\left(\boldsymbol{x}_{t,i}^{sim}\middle|\boldsymbol{x}_{t-L,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim} : \boldsymbol{\theta}\right), \tag{3.7}$$

[2]As also alluded to in Section 2.1.3.3, the samples need not all be drawn from a single Monte Carlo replication and may instead be drawn from the statistical equilibria reached by each replication in an ensemble generated using various random seeds. In practice, we simulate an ensemble of R such Monte Carlo replications for each candidate set of $\boldsymbol{\theta}$ values and combine the samples from each replication into a single random sample.

[3]Note that we have assumed, as in the case of the simulated data, that the empirically-sampled data fluctuates around a stationary level.

for all $L < t \leq T$, where $\boldsymbol{\phi} = \boldsymbol{\phi}(\boldsymbol{\theta})$ are parameters associated with the density estimation procedure.

In our context, we make use of a mixture density network (MDN), a neural network-based approach[4] to conditional density estimation introduced by Bishop (1994). The aforementioned scheme consists of two primary components[5], a mixture of K Gaussian random variables,

$$\tilde{f}\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\phi}\right) = \sum_{k=1}^{K} \alpha_k\left(\boldsymbol{x}\right) \mathcal{N}\left(\boldsymbol{y} \middle| \boldsymbol{\mu}_k\left(\boldsymbol{x}\right), \boldsymbol{\Sigma}_k\left(\boldsymbol{x}\right)\right), \tag{3.8}$$

where we denote $\boldsymbol{x}_{t,i}^{sim}$ by $\boldsymbol{y}$ and $\boldsymbol{x}_{t-L,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim}$ by $\boldsymbol{x}$, and functions α_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ of $\boldsymbol{x}$ which determine the mixture parameters. Here, α_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the outputs of a feedforward neural network taking $\boldsymbol{x}$ as input and having weights and biases $\boldsymbol{\phi}(\boldsymbol{\theta})$, which are determined by training the network on an ensemble of R Monte Carlo replications simulated by the candidate model for parameter set $\boldsymbol{\theta}$. Using the trained MDN, it is then possible to approximate the likelihood of the empirically-sampled data for a given value of $\boldsymbol{\theta}$ as follows[6]:

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{t=1}^{T-L} \tilde{f}(\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+L-1}, \boldsymbol{x}_{t+L}, \boldsymbol{\phi}). \tag{3.9}$$

While alternative density estimation procedures could potentially have been employed, our consideration of MDNs is motivated primarily by their desirable properties. Specifically, MDNs are, in theory, capable of approximating fairly complex conditional distributions. This follows directly from the fact that mixtures of normal random variables are universal density approximators for sufficiently large K (Scott 2015) and the fact that neural networks are universal function approximators (Hornik et al. 1989), provided they are sufficiently expressive. Therefore, provided that K is sufficiently large and the constructed neural network sufficiently deep (and wide), the above methodology should result in accurate conditional density estimates. Additionally, a number of empirical studies, including those of Sugiyama et al. (2012) and Rothfuss et al. (2019), have found MDNs to produce consistently superior performance when compared to a number of prominent alternatives, further motivating their consideration.

3.1.4 Method Comparison and Benchmarking

Given that we have now described our proposed estimation methodology, we proceed to discuss our strategy for benchmarking it against the approach of Grazzini et al. (2017).

[4]It is worth mentioning that a variety of methods in the ABC tradition also make use of neural networks (see, for example, Sheehan and Song (2016) and Papamakarios and Murray (2016)) and thus may share superficial similarities with our proposed methodology. In almost all cases, however, such ABC methods attempt to learn a direct mapping or regression from $\boldsymbol{X}$ to $\boldsymbol{\theta}$ or vice versa. Since, in the case of time series or panel data, $\boldsymbol{X}$ tends to be high-dimensional (at least T in length), it is then required that $\boldsymbol{X}$ first be transformed to a set of summary statistics to yield a tractable density estimation or regression problem, as alluded to in Section 1.1.3.4. Our methodology is thus distinct from the aforementioned approaches by instead considering the more tractable problem of learning the density of $\boldsymbol{x}_{t,i}^{sim}$ conditional on $\boldsymbol{x}_{t-L,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim}$ and thus being able to consider time series or panel data directly without the need to resort to summary statistics.

[5]Note that these discussions are primarily illustrative and serve to briefly describe and motivate our approach. A detailed technical description of its implementation is provided in Appendix 3.A.

[6]This follows directly from the chain rule for probability and our assumption that $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})$ depends only on the past L realised values.

In essence, we follow a practically identical strategy to that employed in Chapter 2 and calculate a Euclidean distance-based loss function. Note, however, that a subtle change is introduced. In this chapter, we ensure that both $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}_{true}$ are normalised to take values in the interval $[0, 1]$ before the loss function value is calculated. This is because, unlike in Chapter 2, several of the considered models involve parameters which are associated with different scales of measurement. Indeed, even relatively small estimation errors associated with parameters that typically take on larger values will increase the loss function value substantially more than relatively large estimation errors associated with parameters that typically take on smaller values if no normalisation is performed. Therefore, for each free parameter, $\theta_j \in [a, b]$, we set

$$\hat{\theta}_j^{[0,1]} = \frac{\hat{\theta}_j - a}{b - a}, \tag{3.10}$$

with an analogous transformation being applied to θ_j^{true}.

As in Chapter 2, the constructed loss function allows us to develop a series of benchmarking exercises in which we compare the loss function values associated with our proposed method and that of Grazzini et al. (2017) for a number of different models, free parameter sets, and $\boldsymbol{\theta}_{true}$ values[7].

In these comparative exercises, we aim to ensure that the overall conditions of the experiments are consistent, regardless of the method used to approximate the likelihood. Therefore, in all cases, we set the length of the proxy for real data to be $T_{emp} = 1000$, the number of Monte Carlo replications in the simulated ensembles to be $R = 100$, the length of each series in the simulated ensembles to be $T_{sim} = 1000$, and the priors for all free parameters to be uniform over the explored parameter ranges, unless stated otherwise. Additionally, we have also used the same lag length, $L = 3$, for all estimation attempts involving our neural network-based method. While seemingly arbitrary, this choice has motivations that are discussed in detail in Section 3.4.1.

Finally, the MCMC algorithm used to sample the posterior and its associated hyperparameters remain unchanged in most experiments and the algorithm is described in detail in Section 2.1.3.4. In general, we set $S = 5000$ and $N = 70$ unless stated otherwise, with convergence typically observed at some point before $s = 1500$, leading us to discard the first 1500 sets as part of a burn-in period[8]. When constructing the posterior samples, we repeat this entire sampling process 5 times and collect the obtained sets to form a larger collection of $5 \times 3500 \times 70 = 1225000$ samples.

3.2 Candidate Models

With our estimation and benchmarking strategies now described, we introduce the candidate models that we attempt to estimate. As in Chapter 2, their selection is primarily justified by their ubiquity, with each having appeared in a number of calibration, estimation, and model selection studies.

[7] While the constructed loss function will act as our primary metric, we will also consider a number of other relevant criteria, such as the standard deviation of the obtained posteriors.

[8] The only exceptions are the experiments involving the Brock and Hommes (1998) model, where we instead consider $S = 10000$ and a burn-in period of 5000, and the empirical exercises in Section 3.3.2, where we instead consider $S = 15000$ and a burn-in period of 10000.

3.2.1 Revisited Models

In addition to a number of new models, we also revisit some of the more interesting models considered in Chapter 2, specifically the Brock and Hommes (1998) model and a random walk capable of replicating simple structural breaks. Both models are described in detail in Section 2.1.2.

3.2.2 Franke and Westerhoff (2012) Model

The next model we discuss shares a number of conceptual similarities with the previously described Brock and Hommes (1998) model, being a heterogeneous agent model that simulates the interactions of traders following a number of trading strategies. It is, however, different in a number of key areas, particularly in how the probability of an agent switching from one strategy to another is determined and in its incorporation of only two trader types, chartists and fundamentalists.

As in the case of the Brock and Hommes (1998) model, the core elements of the model can be expressed as a system of coupled equations

$$p_t = p_{t-1} + \mu \left(n^f_{t-1} d^f_{t-1} + n^c_{t-1} d^c_{t-1} \right), \tag{3.11}$$

$$d^f_t = \phi(p^* - p_t) + \epsilon^f_t,\ \epsilon^f_t \sim \mathcal{N}(0, \sigma^2_f), \tag{3.12}$$

$$d^c_t = \chi(p_t - p_{t-1}) + \epsilon^c_t,\ \epsilon^c_t \sim \mathcal{N}(0, \sigma^2_c), \tag{3.13}$$

$$n^f_t = \frac{1}{1 + \exp(-\beta a_{t-1})}, \tag{3.14}$$

$$n^c_t = 1 - n^f_t, \tag{3.15}$$

where p_t is the log asset price at time t, p^* is the log of the (constant) fundamental value, n^f_t and n^c_t are the market fractions of fundamentalists and chartists respectively at time t, d^f_t and d^c_t are the corresponding average demands, and the remaining symbols all correspond to positive-valued parameters.

At this point, it is worth pointing out that Franke and Westerhoff (2012) do not introduce a single model, but rather a family of related formulations built on the same foundation[9] (Eqns. 3.11-3.15). These models differ in how they define a_t, the attractiveness of fundamentalism relative to chartism at the end of period t, and incorporate a number of different mechanisms, including wealth, herding and price misalignment. This makes the consideration of multiple versions of the model worthwhile and we thus consider two of the proposed versions[10]:

$$a_t = \alpha_n(n^f_t - n^c_t) + \alpha_0 + \alpha_p(p_t - p^*)^2, \tag{3.16}$$

referred to as herding, predisposition and misalignment (HPM), and

$$g^s_t = [\exp(p_t) - \exp(p_{t-1})]\, d^s_{t-2},\ s = \{f, c\}, \tag{3.17}$$

$$w^s_t = \eta w^s_{t-1} + (1 - \eta) g^s_t, \tag{3.18}$$

$$a_t = \alpha_w(w^f_t - w^c_t) + \alpha_0, \tag{3.19}$$

[9] Here we consider only discrete choice approach (DCA) variants of the model, since these were generally found to deliver superior performance by Franke and Westerhoff (2012). The interested reader should refer to the original comparative study for more details on the key differences between DCA and the competing transition probability approach (TPA).

[10] α_n, α_w, and α_p are strictly positive, while α_0 may take on any real value and $\eta \in [0, 1]$.

referred to as wealth and predisposition (WP).

As a final remark, we consider $r_t = p_t - p_{t-1}$, the log-return process, rather than p_t in our estimation attempts.

3.2.3 AR(2)-GARCH(1, 1) Model

In addition to the three main models that we consider in our method comparison study, Section 3.3.2 also provides supplementary exercises involving the application of our proposed methodology to a standard ARMA-GARCH variant. In this case, the model functions primarily as a benchmark against which to compare the performance of the Franke and Westerhoff (2012) model and hence forms part of an elementary model selection investigation, the details of which are elaborated upon in subsequent sections.

Variants of the ARMA-GARCH framework remain ubiquitous in econometric and financial literature and are, despite being conceptually simple, capable of replicating a number of important features of financial returns, such as volatility clustering, ultimately motivating their inclusion as a benchmark. For the purposes of this investigation, we make use of an AR(2)-GARCH(1, 1) model, defined by

$$r_{t+1} = a_1 r_t + a_2 r_{t-1} + \epsilon_{t+1}, \tag{3.20}$$

$$\sigma_{t+1}^2 = \omega + \alpha_1 \epsilon_t^2 + \beta_1 \sigma_t^2, \tag{3.21}$$

$$\epsilon_t = \sigma_t z_t, \tag{3.22}$$

where $\omega > 0$, $\alpha_1, \beta_1 \geq 0$, and $z_t \sim \mathcal{N}(0, 1)$.

3.3 Results and Discussion

We now proceed with the presentation of the results of a comprehensive set of estimation experiments designed to assess the estimation performance of our proposed methodology. These include a set of comparative exercises contrasting its performance with that of the approach of Grazzini et al. (2017) and an additional set of exercises applying it to a number of empirical examples.

3.3.1 Comparative Exercises

In our comparative exercises, we primarily follow the approach laid out in Chapter 2, in which a subset of each model's parameters is estimated using different methods in order to determine loss function values that facilitate direct comparison, as outlined in Section 3.1.4. All parameters not estimated are simply set to their true values (those used to generate the pseudo-empirical time series to which the models are calibrated) when generating the data required to construct a likelihood approximation.

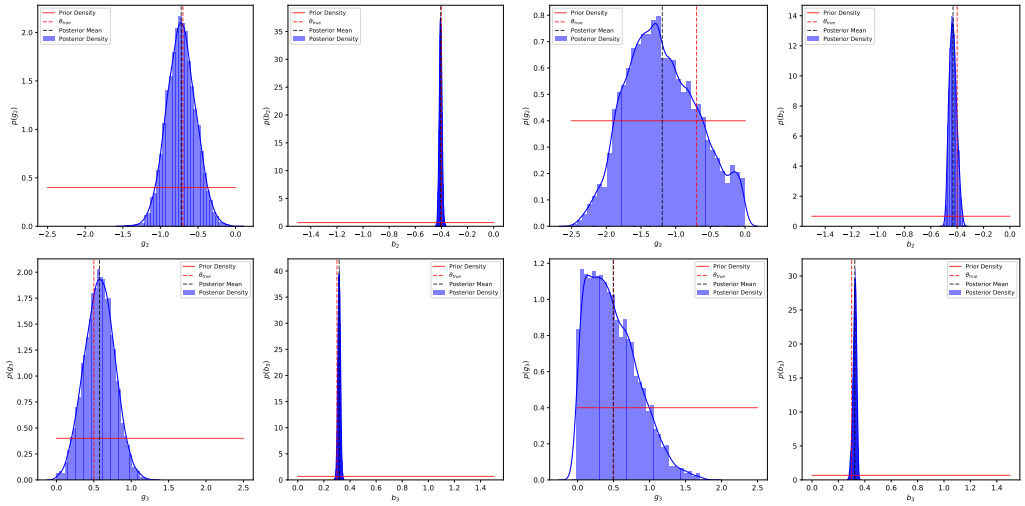
The consideration of parameter subsets is necessitated on a number of grounds. Firstly, the use of such subsets allows us to consider a wide variety of interesting models, while maintaining a level of computational tractability. Secondly, the consideration of additional parameters can, in some models, introduce parameter identification difficulties that stem from collinear or similar relationships between model parameters that render them unidentifiable regardless of the estimation method employed. As an example, a similar protocol is used by Lux (2018) when estimating the Alfarano et al. (2008) model, for which only a subset of the model parameters can be estimated due to the existence of such relationships.

3.3.1.1 Brock and Hommes (1998) Model

We now begin our discussion of the obtained results by focusing on the Brock and Hommes (1998) model[11].

In these experiments, we again, as in Chapter 2, consider a market with $H = 4$ trading strategies and focus on estimating g_2, b_2, g_3, and b_3, the trend following and bias components for two of these strategies. For the first free parameter set, we consider $g_2 \in [-2.5, 0]$, $b_2 \in [-1.5, 0]$, $g_3 \in [0, 2.5]$, and $b_3 \in [0, 1.5]$, corresponding to a contrarian strategy with a negative bias and a trend following strategy with a positive bias respectively. For the second free parameter set, we instead consider $g_2, g_3 \in [0, 2.5]$, $b_2 \in [0, 1.5]$, and $b_3 \in [-1.5, 0]$, corresponding to trend following strategies with positive and negatives biases respectively.

At this point, it is imperative that a number of important features of the Brock and Hommes (1998) model are highlighted, particularly as they relate to our chosen parameter ranges. To elaborate, both here and in Chapter 2, we consider intervals that are restricted a-priori to either positive or negative values. Of course, any of the considered parameters could, in theory, be allowed to take on both positive and negative values and this may lead one to ask why this is not the case in our investigation. While simply stating such assumptions without justification may have been appropriate when considering the model as a small part of a much larger comparative study involving many methodologies in Chapter 2, our more through examination in this chapter necessitates more convincing justifications. We therefore proceed by discussing this issue in more detail and explaining why such restrictions are in fact essential.



(a) MDN (b) KDE

Figure 3.1: Marginal posterior distributions for free parameter set 1 of the Brock and Hommes (1998) model.

From Eqns. 2.6-2.8, it should be apparent that the model makes no distinction between individual trend following (g_1, g_2, g_3, and g_4) and bias (b_1, b_2, b_3, and b_4) parameters;

[11] From this point onwards, we use KDE to refer to the method of Grazzini et al. (2017) and MDN to refer to our proposed method in all tables and figures.

each is subject to identical calculations and plays the same role in each equation. It thus follows that it is the addition of constraints on the values of each of these parameters that defines a given strategy and such constraints are an intrinsic part of the model definition[12]. Therefore, our considered parameter ranges may be seen as defining two distinct variants of the model for which two of the four strategy types differ, as discussed when introducing them above.

Table 3.1: Estimation Result Summary for the Brock and Hommes (1998) Model

	g_2	b_2	g_3	b_3
Param Set 1				
$\boldsymbol{\theta}_{true}$	-0.7	-0.4	0.5	0.3
MDN				
$\boldsymbol{\mu}_{posterior}$	-0.7230	-0.4058	0.5755	0.3170
$\boldsymbol{\sigma}_{posterior}$	0.1904	0.0106	0.1994	0.0099
$\boldsymbol{\sigma}_{sampling}$	0.0054	0.0002	0.0031	0.0002
LS		**0.0338**		
KDE				
$\boldsymbol{\mu}_{posterior}$	-1.1932	-0.4312	0.4963	0.3239
$\boldsymbol{\sigma}_{posterior}$	0.5143	0.0272	0.3434	0.0140
$\boldsymbol{\sigma}_{sampling}$	0.0121	0.0006	0.0094	0.0003
LS		**0.1990**		
Param Set 2				
$\boldsymbol{\theta}_{true}$	0.6	0.65	0.7	-0.55
MDN				
$\boldsymbol{\mu}_{posterior}$	0.5754	0.6460	0.7504	-0.5136
$\boldsymbol{\sigma}_{posterior}$	0.0180	0.0066	0.0153	0.0053
$\boldsymbol{\sigma}_{sampling}$	0.0030	0.0013	0.0030	0.0005
LS		**0.0332**		
KDE				
$\boldsymbol{\mu}_{posterior}$	0.4960	0.6861	0.7252	-0.5521
$\boldsymbol{\sigma}_{posterior}$	0.0570	0.0231	0.0215	0.0049
$\boldsymbol{\sigma}_{sampling}$	0.0021	0.0008	0.0003	0.0002
LS		**0.0491**		

$g_1 = b_1 = b_4 = 0$, $g_4 = 1.01$, $r = 0.01$, $\beta = p^* = 10$, and $\sigma = 0.04$ for both free parameter sets.

The above features also have important consequences for the estimation of the model in general. To illustrate, the model output would be identical if, for example, we considered two different model configurations, $\{g_2 = -0.7, b_2 = -0.4, g_3 = 0.5, b_3 = 0.3\}$ and $\{g_2 = 0.5, b_2 = 0.3, g_3 = -0.7, b_3 = -0.4\}$, with the values of the remaining parameters being identical. This is because the model is generally agnostic to differences in the value of the index h to which a particular strategy is assigned, provided that the overall composition of

[12]While the strategies corresponding to positive and negative values of g_h and b_h are self-explanatory, it is worth pointing out that setting $g_h = 0$ and $b_h = 0$ would result in a fundamentalist strategy.

strategies remains unchanged, as is the case for the two example model configurations given above. This implies that if we were to instead consider symmetric parameter ranges, such as $g_2, g_3 \in [-2.5, 2.5]$, and $b_2, b_3 \in [-1.5, 1.5]$ for the first free parameter set, the considered free parameters would become unidentifiable, regardless of the estimation approach employed. It is therefore vital that we constrain the explored parameter ranges to ensure that the model and corresponding estimation problem are well-specified and that the associated strategies are distinct[13].

Now, referring to Figure 3.1, which presents a graphical illustration of the estimation results associated with the first free parameter set, we observe that there are a number of key differences in performance that emerge between our proposed methodology and that of Grazzini et al. (2017). While both approaches seem to perform somewhat comparably when estimating b_2, g_3, and b_3, producing posterior means within reasonably close proximity to the corresponding true parameter values, more significant differences emerge when considering g_2. Specifically, we see that while the posterior mean associated with the method of Grazzini et al. (2017) is a relatively poor estimate, our proposed methodology fares far better and maintains a consistent level of performance. Additionally, we also observe that the posteriors associated with our proposed methodology are significantly narrower and more peaked, with their density concentrated in a smaller region of the parameter space. This can be seen as indicative of reduced estimation uncertainty.

Table 3.1 elaborates on these findings and reveals that similar behaviours also emerge in the case of the second free parameter set. Specifically, we find that differences in estimation performance are generally more pronounced when considering the posterior means for g_2 as opposed to those associated with g_3, b_2, and b_3. Since our proposed methodology produces more accurate g_2 estimates, we ultimately observe lower loss function values for both free parameter sets. We also observe that our approach results in reduced posterior standard deviations ($\boldsymbol{\sigma}_{posterior}$) consistently for all but one of the free parameters[14], in line with our observation of reduced estimation uncertainty in Figure 3.1. It can thus be concluded that our proposed methodology results in meaningful improvements in estimation performance in the context of the Brock and Hommes (1998) model.

Finally, in Section 3.1.4, where we describe our experimental setup, we indicate that we run the sampling procedure multiple times from different starting points in the parameter space and combine the obtained samples into a single, larger sample from which we estimate $\boldsymbol{\mu}_{posterior}$ and $\boldsymbol{\sigma}_{posterior}$. We can, however, estimate the posterior mean for each of these runs individually and determine the standard deviation of $\boldsymbol{\mu}_{posterior}$ across the instantiations of the algorithm, which we call $\boldsymbol{\sigma}_{sampling}$. As shown in Table 3.1, this standard deviation is generally very small for both methods, suggesting that the posterior mean estimates are robust to changes in the initial conditions of the MCMC algorithm[15].

3.3.1.2 Random Walks with Structural Breaks

Moving on from the Brock and Hommes (1998) model, we now discuss the estimation of a random walk incorporating a structural break. In these experiments, we consider a fixed

[13]The interested reader should refer to the frequentist study performed by Kukacka and Barunik (2017), where the necessity of such constraints is discussed in more detail. Additionally, one may also wish to refer to Appendix 3.B, where we provide additional experiments involving the relaxation of some of the aforementioned constraints.

[14]This refers to the posterior standard deviation obtained for b_3 in the second free parameter set, with the difference between the methodologies being negligible in this case.

[15]This is true for all free parameter sets and models considered in this investigation.

structural break location[16], $\tau = 700$, and determine the extent to which both methods are capable of estimating the pre- and post-break drift, $d_1, d_2 \in [-2, 2]$, and volatility, $\sigma_1, \sigma_2 \in [0, 10]$, for differing underlying changes in the dynamics. While the loss function described in Section 3.1.4 will still be used as our primary metric, we note that since the considered free parameters directly define the dynamics that characterise the different regimes of the data, it would also be worthwhile to assess the extent to which the competing approaches are able to correctly identify the relationships between the parameters and hence the shift in the pre- and post-break dynamics (Δ_d and Δ_σ).

Table 3.2: Estimation Result Summary for the Random Walk Model (Increasing Volatility)

	σ_1	σ_2	Δ_σ
Param Set 1			
$\boldsymbol{\theta}_{true}$	1	2	1
MDN			
$\boldsymbol{\mu}_{posterior}$	0.9478	1.9560	1.0082
$\boldsymbol{\sigma}_{posterior}$	0.0337	0.1015	–
$\boldsymbol{\sigma}_{sampling}$	0.0007	0.0017	–
LS		**0.0068**	
KDE			
$\boldsymbol{\mu}_{posterior}$	0.9684	1.9040	0.9354
$\boldsymbol{\sigma}_{posterior}$	0.0383	0.0644	–
$\boldsymbol{\sigma}_{sampling}$	0.0009	0.0019	–
LS		**0.0101**	
Param Set 2			
$\boldsymbol{\theta}_{true}$	1	2	1
MDN			
$\boldsymbol{\mu}_{posterior}$	0.9468	1.9531	1.0063
$\boldsymbol{\sigma}_{posterior}$	0.0336	0.1034	–
$\boldsymbol{\sigma}_{sampling}$	0.0013	0.0016	–
LS		**0.0071**	
KDE			
$\boldsymbol{\mu}_{posterior}$	0.9684	1.9000	0.9316
$\boldsymbol{\sigma}_{posterior}$	0.0393	0.0594	–
$\boldsymbol{\sigma}_{sampling}$	0.0006	0.0020	–
LS		**0.0104**	

$d_1 = 0.4$ and $d_2 = 0.5$ for free parameter set 1 and $d_1 = 0.1$ and $d_2 = 0.2$ for free parameter set 2.

Before proceeding, however, there are a number of nuances that should be highlighted. Recall from Appendix 2.A that the model produces non-stationary time series and therefore violates a key assumption of the method of Grazzini et al. (2017). For this reason, it is

[16]This induces a degree of asymmetry in the data and results in a more challenging and realistic estimation problem than $\tau = 500$.

necessary to once again consider the series of first differences, $x_t - x_{t-1}$, rather than x_t itself. While our approach does not make stationarity assumptions, we have none the less considered the series of first differences when applying both methods to make the comparison as fair as possible. It should also be noted that, as implied in Chapter 2, we have assumed the location of the structural break to be unknown or difficult to determine a-priori (as is the case in most practical problems), meaning that we apply both estimation approaches to the full time series data to estimate both the pre- and post-break parameters simultaneously. If, however, the location of the structural break was known, it would be possible to estimate the relevant parameters separately using appropriate subsets of the data, a less challenging undertaking that we do not consider here.

Table 3.3: Estimation Result Summary for the Random Walk Model (Increasing Drift)

	d_1	d_2	Δ_d
Param Set 3			
$\boldsymbol{\theta}_{true}$	0.4	0.5	0.1
MDN			
$\boldsymbol{\mu}_{posterior}$	0.4871	0.5442	0.0571
$\boldsymbol{\sigma}_{posterior}$	0.0397	0.1121	–
$\boldsymbol{\sigma}_{sampling}$	0.0008	0.0034	–
LS		**0.0244**	
KDE			
$\boldsymbol{\mu}_{posterior}$	0.5196	0.3206	−0.1991
$\boldsymbol{\sigma}_{posterior}$	0.0520	0.1512	–
$\boldsymbol{\sigma}_{sampling}$	0.0009	0.0034	–
LS		**0.0539**	
Param Set 4			
$\boldsymbol{\theta}_{true}$	0.4	0.7	0.3
MDN			
$\boldsymbol{\mu}_{posterior}$	0.5046	0.6913	0.1866
$\boldsymbol{\sigma}_{posterior}$	0.0403	0.1126	–
$\boldsymbol{\sigma}_{sampling}$	0.0009	0.0020	–
LS		**0.0263**	
KDE			
$\boldsymbol{\mu}_{posterior}$	0.5312	0.5017	−0.0296
$\boldsymbol{\sigma}_{posterior}$	0.0528	0.1453	–
$\boldsymbol{\sigma}_{sampling}$	0.0016	0.0057	–
LS		**0.0595**	

$\sigma_1 = 1$ and $\sigma_2 = 2$ for both free parameter sets.

Now, referring to Table 3.2, we see that both our proposed estimation methodology and that of Grazzini et al. (2017) perform similarly well when attempting to estimate the pre- and post-break volatility, with both producing reasonable estimates for the free parameters and both being able to identity the correct shift in the dynamics. Referring to Tables 3.3 and 3.4,

however, we see that more pronounced differences emerge when attempting to estimate the pre- and post-break drift. While this is clearly evident from the fact that the loss function values associated with our proposed methodology are noticeably lower in all cases, a more detailed analysis reveals further distinctions worth mentioning. Table 3.3, which presents the results for cases involving an increasing drift, reveals that our proposed methodology has correctly identified an increasing trend in both cases and has also more or less correctly identified that the increase in drift for parameter set 4 is three times that of parameter set 3. In contrast to this, the method of Grazzini et al. (2017) incorrectly suggests a decreasing trend in both cases. Table 3.4, which presents the results for cases involving a decreasing drift, similarly shows that our proposed methodology delivers superior performance when attempting to identify the change in drift.

Table 3.4: Estimation Result Summary for the Random Walk Model (Decreasing Drift)

	d_1	d_2	Δ_d
Param Set 5			
$\boldsymbol{\theta}_{true}$	0.5	0.4	-0.1
MDN			
$\boldsymbol{\mu}_{posterior}$	0.5709	0.4741	-0.0968
$\boldsymbol{\sigma}_{posterior}$	0.0406	0.1329	–
$\boldsymbol{\sigma}_{sampling}$	0.0013	0.0031	–
LS		**0.0256**	
KDE			
$\boldsymbol{\mu}_{posterior}$	0.6033	0.2511	-0.3523
$\boldsymbol{\sigma}_{posterior}$	0.0507	0.1493	–
$\boldsymbol{\sigma}_{sampling}$	0.0008	0.0031	–
LS		**0.0453**	
Param Set 6			
$\boldsymbol{\theta}_{true}$	0.7	0.4	-0.3
MDN			
$\boldsymbol{\mu}_{posterior}$	0.7605	0.4444	-0.3161
$\boldsymbol{\sigma}_{posterior}$	0.0409	0.1524	–
$\boldsymbol{\sigma}_{sampling}$	0.0009	0.0050	–
LS		**0.0188**	
KDE			
$\boldsymbol{\mu}_{posterior}$	0.7863	0.2852	-0.5010
$\boldsymbol{\sigma}_{posterior}$	0.0521	0.1494	–
$\boldsymbol{\sigma}_{sampling}$	0.0012	0.0046	–
LS		**0.0359**	

$\sigma_1 = 1$ and $\sigma_2 = 2$ for both free parameter sets.

This change in the relative performances of each method when estimating the drift rather than the volatility is a direct consequence of the relationship between the deterministic and stochastic components of the model. For the selected parameter ranges, the random

fluctuations, ϵ_t, dominate the evolution of the model, with the drift producing a more subtle effect, particularly after the structural break occurs. For this reason, correctly estimating the pre- and post-break volatility is a far less challenging task than estimating the pre- and post-break drift. Therefore, while both methods perform well when estimating parameters associated with dominant effects like volatility, our method's incorporation of dependence on previously observed values seems to be important when estimating parameters related to more nuanced and less dominant aspects of a model.

3.3.1.3 Franke and Westerhoff (2012) Model

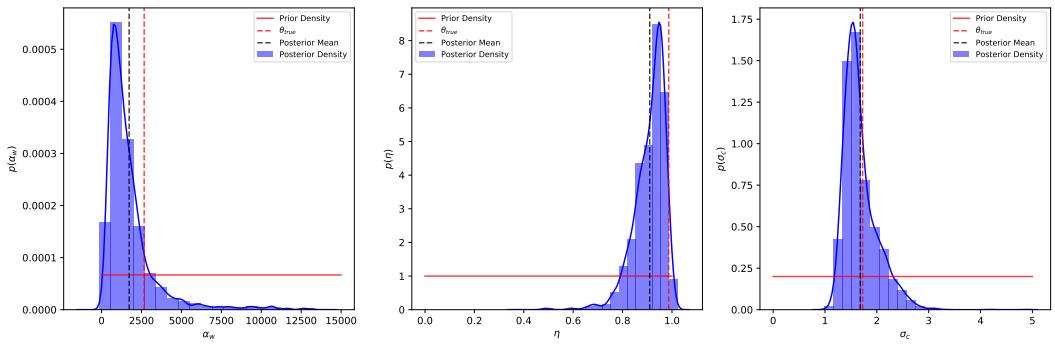
Table 3.5: Estimation Result Summary for the Franke and Westerhoff (2012) Model

	α_0	α_n	α_p	σ_c
Param Set HPM				
$\boldsymbol{\theta}_{true}$	-0.327	1.79	18.43	2.087
MDN				
$\boldsymbol{\mu}_{posterior}$	-0.1698	1.9258	17.3057	2.2741
$\boldsymbol{\sigma}_{posterior}$	0.0752	0.0635	2.1941	0.2608
$\boldsymbol{\sigma}_{sampling}$	0.0065	0.0032	0.12037	0.0035
LS		**0.1239**		
KDE				
$\boldsymbol{\mu}_{posterior}$	-0.1366	1.8564	16.3857	2.2733
$\boldsymbol{\sigma}_{posterior}$	0.0941	0.1315	2.8208	0.4220
$\boldsymbol{\sigma}_{sampling}$	0.0086	0.00751	0.2437	0.0311
LS		**0.1483**		
	α_w	η	σ_c	
Param Set WP				
$\boldsymbol{\theta}_{true}$	2668	0.987	1.726	
MDN				
$\boldsymbol{\mu}_{posterior}$	1728.3105	0.9099	1.6799	
$\boldsymbol{\sigma}_{posterior}$	1648.4308	0.0628	0.3240	
$\boldsymbol{\sigma}_{sampling}$	241.6884	0.0021	0.0799	
LS		**0.0997**		
KDE				
$\boldsymbol{\mu}_{posterior}$	2207.8908	0.6327	1.4248	
$\boldsymbol{\sigma}_{posterior}$	2569.5501	0.2688	0.1796	
$\boldsymbol{\sigma}_{sampling}$	351.1881	0.0180	0.0304	
LS		**0.3607**		

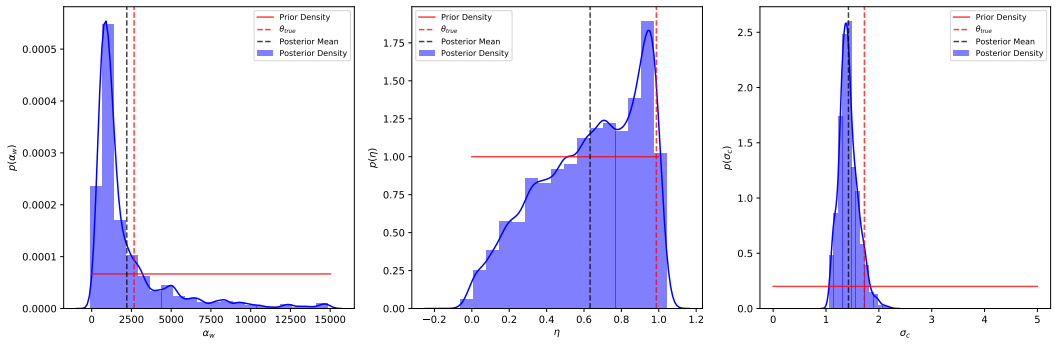
$\mu = 0.01$, $\beta = 1$, $\phi = 0.12$, $\chi = 1.5$, and $\sigma_f = 0.758$ for the HPM parameter set and $\mu = 0.01$, $\beta = 1$, $\phi = 1$, $\chi = 0.9$, $\alpha_0 = 2.1$, and $\sigma_f = 0.752$ for the WP parameter set, as suggested by Franke and Westerhoff (2012).

As stated in Section 3.2.2, the final model we consider has a number of alternate configurations differing in how the attractiveness of fundamentalism relative to chartism, a_t, is

determined during each period. For this reason, we consider two of these configurations, HPM and WP, and focus on estimating the parameters associated with the rules governing a_t: $\alpha_n \in [0, 2]$, $\alpha_0 \in [-1, 1]$, $\alpha_p \in [0, 20]$, $\alpha_w \in [0, 15000]$, and $\eta \in [0, 1]$, while also estimating the standard deviation of the noise term appearing in the chartist demand equation, $\sigma_c \in [0, 5]$. Our selected parameter ranges take the intervals presented in Table 1 of Barde (2016) as a basis, with only slight modifications being made for the purposes of our investigation[17].



(a) MDN



(b) KDE

Figure 3.2: Marginal posterior distributions for the WP parameter set of the Franke and Westerhoff (2012) model.

Referring to Table 3.5, we see that our proposed estimation methodology appears slightly more effective than that of Grazzini et al. (2017) for the HPM parameter set, producing competitive estimates for all of the considered free parameters and resulting in a lower loss function value. Nevertheless, the estimates do not differ substantially when comparing the methods. Despite this, we see, in what is a seemingly analogous trend to what was observed in the random walk experiments, that the differences in performance are more pronounced for the WP parameter set. In particular, we see a substantial difference in the loss function values associated with each method, brought about by differences in the quality of estimates produced for η.

[17] As an example, we extend the length of certain intervals (α_0 and α_p) and slightly shift others to begin at 0 while maintaining an identical length (α_n and σ_c).

As illustrated in Figure 3.2, the method of Grazzini et al. (2017) produces a wide posterior for η that is dispersed across the entirety of the explored parameter range, which results in a relatively poor estimate. In contrast to this, we see that the proposed methodology fares better, producing a far narrower posterior and a significantly more accurate estimate. While it is nontrivial to identify any definitive causes for the observed behaviours due to the nonlinear nature of heterogeneous agent models, it is worth pointing out that the inclusion of wealth dynamics in the WP version of the model introduces a dependence of a_t on the previous return via Eqns. 3.17-3.19, which may in turn increase the strength of the relationship between the current and previously observed values in the log-return time series.

As a final remark, notice that for the vast majority of the free parameters considered, the proposed methodology also results in lower posterior standard deviations, as was the case for the previously considered models.

3.3.1.4 Overall Summary

In the preceding subsections, we have focused primarily on analysing the results on a case-by-case basis. Here, however, we provide a summative comparison across all of the considered models. This is achieved though the consideration of a number of key performance metrics, presented in Table 3.6, which compare the approaches at both a global and individual parameter level.

The first of the aforementioned metrics, and the most important, $LS_{mdn} < LS_{kde}$, indicates how often the proposed methodology results in lower loss function values, and hence measures its relative ability to recover the true parameter set. We observe that in all cases considered, our methodology results in lower loss function values, which can be seen as indicative of dominance at the global level.

Table 3.6: Estimation Result Summary Across All Models

Outcome	Percentage of Cases
$LS_{mdn} < LS_{kde}$	100
$\lvert\mu^i_{mdn} - \theta^i_{true}\rvert < \lvert\mu^i_{kde} - \theta^i_{true}\rvert$	70.37
$\sigma^i_{mdn} < \sigma^i_{kde}$	81.48

The second metric, $|\mu^i_{mdn} - \theta^i_{true}| < |\mu^i_{kde} - \theta^i_{true}|$, determines how often our proposed methodology produces superior estimates for individual parameters. In some situations, one might find that the estimates obtained for a subset of the free parameters by the method of Grazzini et al. (2017) are superior, even if the overall estimate for the entire free parameter set is not as good. Nevertheless, we find that in over 70% of cases, our methodology also results in superior estimates at the level of individual parameters, a comfortable majority. It should also be noted that in almost all situations where $|\mu^i_{mdn} - \theta^i_{true}| > |\mu^i_{kde} - \theta^i_{true}|$, such as σ_1 in the random walk model and σ_c in the Franke and Westerhoff (2012) model (HPM parameter set), the differences in the estimates produced by both methods are incredibly small. In contrast to this, a sizeable number of cases where $|\mu^i_{mdn} - \theta^i_{true}| < |\mu^i_{kde} - \theta^i_{true}|$, such as g_2 in the first free parameter set of the Brock and Hommes (1998) model and η in the second free parameter set of the Franke and Westerhoff (2012) model, are characterised by comparatively large differences in the estimates obtained by the competing approaches.

This suggests that our proposed methodology also demonstrates a degree of dominance at the level of individual parameters.

The final metric, $\sigma^i_{mdn} < \sigma^i_{kde}$, indicates how frequently our proposed methodology results in reduced posterior standard deviations for individual parameters, which can be viewed as roughly quantifying estimation uncertainty. We find that our approach again delivers consistently superior performance along these lines, producing a smaller posterior standard deviation in over 81% of the considered cases.

Based on the evidence presented by the above metrics as a whole, it would appear that our proposed methodology does indeed compare favourably to that of Grazzini et al. (2017), which was itself already shown to dominate a number of other contemporary approaches in the literature in Chapter 2. This ultimately validates our method as a worthwhile addition to the growing toolbox of estimation methods for economic simulation models.

3.3.2 Empirical Examples

In the preceding exercises, we focused primarily on a comparison of the estimation performance of our proposed methodology with that of the approach of Grazzini et al. (2017) in the context of pseudo-empirical data. We now proceed to empirical applications of our approach involving the DCA-HPM variant[18] of the Franke and Westerhoff (2012) model, which we benchmark against a standard and widely-used approach for modelling financial returns, an AR(2)-GARCH(1, 1) model, in an attempt to provide a simple example of the competitiveness of the ABM paradigm relative to other approaches.

3.3.2.1 Full Model Estimation

Unlike in the previously considered cases, we estimate all model parameters simultaneously in each exercise rather than limiting ourselves to parameter subsets. It is therefore imperative that, before attempting to fit both models to empirical data using our proposed methodology, we first demonstrate our approach's ability to recover reasonable values for each of the considered free parameters. Therefore, as in Section 3.3.1, we begin by calibrating both models to pseudo-empirical time series data as a validation step.

To account for the increased number of free parameters considered in these experiments, we are required to adjust certain aspects of our overall procedure. In particular, we increase the length of the pseudo-empirical data and model Monte Carlo replications, T_{emp} and T_{sim} respectively, to 2000, while reducing the number of model Monte Carlo replications[19], R, to 50. Additionally, we also increase the number of sample sets and burning-in period associated with the adaptive Metropolis-Hastings algorithm to 15000 and 10000 respectively.

As before, we employ uniform priors over the considered parameter ranges. For the AR-GARCH model, we assume $a_1, a_2 \in [-1.5, 1.5]$ and ω, α_1, $\beta_1 \in [0, 2]$. For the Franke and Westerhoff (2012) model, we again slightly modify the ranges originally considered in the comparative exercise conducted by Barde (2016), leading us to assume $\phi, \chi \in [0, 4]$, $\alpha_0 \in [-1, 1]$, $\alpha_n \in [0, 2]$, $\alpha_p \in [0, 20]$, $\sigma_f \in [0, 1.25]$, and $\sigma_c \in [0, 5]$. It should be noted that, as pointed out by both Franke and Westerhoff (2012) and Bertschinger and Mozzhorin (2020), μ and β are redundant parameters that simply act as multiplicative constants for

[18] Franke and Westerhoff (2012) find DCA-HPM to be the best performing variant of the model in their moment-based investigation and Barde (2016) similarly finds it to be competitive relative to a number of competing models and paradigms, motivating our choice.

[19] Since the considered value of T_{sim} is double that used in Section 3.3.1, our selected value of R (half that considered in Section 3.3.1) still results in the same MDN training set size.

$\phi, \chi, \sigma_f, \sigma_c$ and $\alpha_0, \alpha_n, \alpha_p$ respectively and should therefore not be estimated, leading us to fix their values to the defaults suggested by Franke and Westerhoff (2012), as indicated in Table 3.7.

Table 3.7: Estimation Result Summary for the Full Franke and Westerhoff (2012) Model on Several Datasets

	ϕ	χ	α_0	α_n	α_p	σ_f	σ_c
Pseudo-empirical Data							
$\boldsymbol{\theta}_{true}$	0.12	1.5	−0.327	1.79	18.43	0.758	2.087
$\boldsymbol{\mu}_{posterior}$	0.0979	1.8558	−0.1793	1.7957	15.5455	0.6953	2.3603
$\boldsymbol{\sigma}_{posterior}$	0.0902	1.1577	0.1149	0.1920	3.3000	0.0436	0.3998
$\boldsymbol{\sigma}_{sampling}$	0.0067	0.0398	0.0168	0.0209	0.2065	0.0027	0.0548
Nikkei 225							
$\boldsymbol{\mu}_{posterior}$	0.4159	1.9692	−0.0702	1.9243	13.6797	0.4188	4.7267
$\boldsymbol{\sigma}_{posterior}$	0.1038	1.1179	0.0225	0.0619	2.8588	0.0914	0.2057
$\boldsymbol{\sigma}_{sampling}$	0.0064	0.0721	0.0021	0.0059	0.1649	0.0078	0.0092
FTSE 100							
$\boldsymbol{\mu}_{posterior}$	0.1255	1.9660	−0.0003	1.7907	15.0406	0.4011	3.139
$\boldsymbol{\sigma}_{posterior}$	0.0676	1.1477	0.0589	0.2024	3.7324	0.0659	0.5914
$\boldsymbol{\sigma}_{sampling}$	0.0160	0.1372	0.0086	0.0339	0.0918	0.011	0.0889

$\mu = 0.01$ and $\beta = 1$, as suggested by Franke and Westerhoff (2012).

Additionally, particular attention should be paid to σ_f, where we consider a more constrained range of variation compared to that of σ_c. This is necessitated primarily due to idiosyncrasies that emerge regardless of the estimation methodology employed. Consistent with the findings of Bertschinger and Mozzhorin (2020), who employ a probabilistic programming approach, we find that the estimation of the DCA-HPM variant of the Franke and Westerhoff (2012) model can, in certain situations, lead to a bimodal posterior that renders certain parameters unidentifiable. When comparing the two obtained modes, it emerges that it is in fact the values of σ_c and σ_f that drive the observed bimodality, with one mode being associated with $\sigma_f < \sigma_c$ and another with $\sigma_f > \sigma_c$[20], despite both modes having very similar log-likelihoods. This necessitates the addition of a constraint that enforces parameter identifiability, leading us to consider a reduced range of variation for σ_f as indicated above, ensuring that the mode for which $\sigma_f < \sigma_c$ is always returned in cases where the posterior is bimodal. As stated by Bertschinger and Mozzhorin (2020), the mode in which $\sigma_f > \sigma_c$ leads to model dynamics in which fundamentalist agents are the primary drivers of volatility. This is in contrast to the original intent of the model, where fundamentalists are expected to instead produce a stabilising effect and chartists are intended to drive volatile price movements, motivating our choice.

Directly following the preceding validation exercise, we apply an identical protocol[21] to the log-return time series associated with the closing prices of two major stock market

[20] In other words, the values of σ_f and σ_c are more or less flipped.

[21] Only a single, subtle change is introduced. In particular, we assume $\omega \in [0, 2\hat{\sigma}^2]$, where $\hat{\sigma}^2$ is the variance of the empirical series. Since ω essentially defines the minimum possible value for the (time-varying) variance of the AR process errors, values of ω much larger than $\hat{\sigma}^2$ would generally result in a simulated variance

Table 3.8: Estimation Result Summary for the Full AR(2)-GARCH(1, 1) Model on Several Datasets

	a_1	a_2	ω	α_1	β_1
Pseudo-empirical Data					
$\boldsymbol{\theta}_{true}$	0.2	0.25	0.1	0.5	0.2
$\boldsymbol{\mu}_{posterior}$	0.2029	0.2713	0.0747	0.6230	0.2207
$\boldsymbol{\sigma}_{posterior}$	0.0233	0.0205	0.0085	0.0495	0.0440
$\boldsymbol{\sigma}_{sampling}$	0.0005	0.0006	0.0004	0.0017	0.0011
Nikkei 225					
$\boldsymbol{\mu}_{posterior}$	−0.0110	0.0475	1.4431	0.101	0.8949
$\boldsymbol{\sigma}_{posterior}$	0.0284	0.0287	0.5356	0.0184	0.0202
$\boldsymbol{\sigma}_{sampling}$	0.0018	0.0017	0.0068	0.0025	0.0028
FTSE 100					
$\boldsymbol{\mu}_{posterior}$	−0.0002	−0.0136	0.967	0.0962	0.8935
$\boldsymbol{\sigma}_{posterior}$	0.0248	0.0281	0.484	0.0235	0.0287
$\boldsymbol{\sigma}_{sampling}$	0.0013	0.0011	0.0040	0.0016	0.0021

All values associated with ω for the empirical datasets have been multiplied by 10^6 for improved readability.

indices, the FTSE 100 from 03-01-2012 to 30-12-2019, and the Nikkei 225 from 01-03-2011 to 30-12-2019, with the date ranges selected to ensure that the resulting empirical time series are both recent and of a similar length to the pseudo-empirical series used in the validation step[22]. The complete sets of estimation results for both the Franke and Westerhoff (2012) and AR-GARCH models are presented in Tables 3.7 and 3.8 respectively, with a number of aspects worth highlighting. To begin, we notice that, for both models, the validation exercises involving simulated data demonstrate that, despite an increase in the considered number of free parameters, the quality of the obtained estimation results remains in line with those associated with the benchmarking exercises presented in Section 3.3.1, with the posterior means of all free parameters being within reasonably close proximity to their true values. This is ultimately very reassuring, and provides some level of confidence in the validity of the results obtained when applying our protocol to empirical data. Proceeding to the empirical results themselves, it should additionally be noted that, unlike the parameters of the AR-GARCH model, the parameters of the Franke and Westerhoff (2012) model are readily interpretable, and may thus provide a number of meaningful insights regarding the

substantially larger than its empirical counterpart and therefore very low log-likelihood values. A similar empirically-derived restriction on the range of a variance parameter is employed in the Bayesian estimation exercise conducted by Lux (2018).

[22]The Nikkei 225 is associated with fewer trading days than the FTSE 100 over the period beginning on 03-01-2012 and ending on 30-12-2019. The consideration of this period would have resulted in the Nikkei 225 dataset having fewer than $T_{emp} = 2000$ observations, which is the length of the pseudo-empirical data used in our validation exercise. Therefore, in order to ensure that the number of observations in the Nikkei 225 dataset is sufficient (equal to or exceeding the number of observations in our original pseudo-empirical dataset), we have slightly extended the considered date range by a few months. This is not dissimilar to the work of Barde (2016) and Kukacka and Barunik (2017), who also consider datasets with slightly different starting dates to account for differing trading conventions and a host of other factors.

considered markets. In particular, we notice that despite very similar estimates for σ_f, the Nikkei dataset results in a noticeably larger estimate for ϕ when compared to that obtained for the FTSE, suggesting that fundamentalist traders in the former market have a greater tendency to buy or sell in response to a given level of perceived mispricing. Additionally, we also observe that despite virtually identical estimates for χ, the Nikkei dataset results in a noticeably larger estimate for σ_c, suggesting greater heterogeneity in the demands of chartist traders in the Japanese market. As a final remark, it is also worth noting that the Nikkei is associated with a stronger herding tendency and a more significant (though still relatively weak) predisposition to chartist trading.

3.3.2.2 Model Comparison

With both models now being calibrated to empirical data, it is relatively straightforward to also provide a simple model comparison exercise. Indeed, given that ABMs are often positioned as more interpretable alternatives to traditional time series models, it is worth considering whether the calibrated Franke and Westerhoff (2012) model is competitive when benchmarked against a more traditional AR-GARCH model. Along these lines, we compare the two candidate models according to two distinct criteria.

Table 3.9: Estimated Model Comparison

	Moment	Empirical	FW	AR-GARCH
Nikkei 225				
	Std. Dev.	0.0131	**0.0138** (0.0001)	0.0165 (0.0007)
	Kurt.	6.1497	4.1761 (0.0933)	**4.2086** (0.3188)
	Skew.	−0.6119	**−0.0259** (0.0178)	−0.0007 (0.0208)
	Abs. ACF 1	0.2065	**0.2771** (0.0030)	0.2988 (0.0070)
	Abs. ACF 3	0.2002	**0.2611** (0.0029)	0.2899 (0.0069)
	Abs. ACF 5	0.1192	**0.2498** (0.0029)	0.2852 (0.0071)
	MIC	–	**12929.9118**	12933.2941
FTSE 100				
	Std. Dev.	0.0084	**0.0084** (0.0001)	0.0093 (0.0001)
	Kurt.	2.1766	**2.0706** (0.0557)	2.6022 (0.1862)
	Skew.	−0.2079	**−0.0073** (0.0120)	−0.0044 (0.0139)
	Abs. ACF 1	0.2197	0.1821 (0.0031)	**0.2405** (0.0060)
	Abs. ACF 3	0.2080	0.1752 (0.0031)	**0.2309** (0.0060)
	Abs. ACF 5	0.1580	**0.1713** (0.0032)	0.2249 (0.0060)
	MIC	–	**10865.7132**	10874.9373

The best performing model is indicated in bold. All values represent the means over 200 model paths with the associated standard errors in brackets. The MIC scores are calculated with a discretisation resolution of $r = 8$, a context length of $L = 3$, and discretisation bounds $[-0.1, 0.1]$, and are associated with the same ensembles of 200 Monte Carlo replications considered in the moment-based comparative exercise.

As a first step, we perform a simple moment-based comparison in the vein of Franke and Westerhoff (2012), where we initialise each model using the final parameter estimates

obtained in our previous exercise ($\boldsymbol{\mu}_{posterior}$ in Tables 3.7 and 3.8) and proceed to generate sets of 200 Monte Carlo replications of the log-return process of length $T_{sim} = 2000$ (one set of replications for each model and empirical series combination, for a total of four). For each of these replications, we estimate the values of 6 moments, which may then be used to determine the means and associated standard errors[23] of these moments for each replication set and hence facilitate an elementary comparative exercise[24]. As with any moment-based investigation, the choice of moments is of critical importance, leading us to select quantities that correspond to well-known return time series stylised facts. In particular, we consider the standard deviation, excess kurtosis, and skewness of the raw return series, as well as the autocorrelation coefficients of the absolute return series at lags 1, 3, and 5. The inclusion of the kurtosis and skewness is motivated chiefly by the fat-tailed and negatively-skewed distributions associated with empirically-observed returns and, along with the standard deviation, is intended to capture key features of the unconditional return distribution. In contrast, the inclusion of the aforementioned autocorrelation coefficients is intended to capture the phenomenon of volatility clustering and thus instead focuses on the conditional distribution and temporal dynamics of the return process.

Referring to Table 3.9, which presents the results of our moment comparison exercise, we find that the Franke and Westerhoff (2012) model generally delivers the best performance, with the mean values of 5 and 4 of the considered moments being closer to their empirical values than the corresponding AR-GARCH equivalents for the Nikkei and FTSE datasets respectively, providing some evidence of the competitiveness of ABMs relative to more traditional approaches. Of course, moment-based model selection techniques, in requiring an arbitrary set of moments or summary statistics to be constructed, suffer from identical weaknesses to those encountered in the estimation context, leading us to keep our discussion of the above results relatively brief and additionally necessitating the consideration of a more principled set of model comparison tools.

Along these lines, we also compare the estimated models to the empirical data using the MIC, which has already been discussed in detail in Chapter 2 from the perspective of its value as an estimation method[25]. With reference to Table 3.9, which also presents the MIC scores associated with each candidate model, we see that the Franke and Westerhoff (2012) model again delivers the best performance, providing further evidence of its competitiveness relative to the more-established AR-GARCH model and reinforcing the findings of Barde (2016) and Bertschinger and Mozzhorin (2020).

Ultimately, the provided comparative exercises are primarily illustrative and included to provide a simple demonstration of the ability of our approach to facilitate model comparison and selection. The preceding exercises could, of course, be extended to a larger set of candidate models and incorporate additional Bayesian model selection techniques, such as cross-validation, to further explore these findings and yield a more comprehensive and robust set of model selection experiments. This is explored in detail in Chapter 4.

[23]The standard error is given by $\frac{\tilde{\sigma}}{\sqrt{N}}$, where $\tilde{\sigma}$ is the sample standard deviation over the ensemble of Monte Carlo replications and N is the number of Monte Carlo replications.

[24]Since our model fitting process is likelihood rather than moment-based, this exercise can, to some extent, be seen as a form of out-of-sample testing.

[25]Although Chapter 2 demonstrates that the MIC is not well-suited to estimation, it still has value when used to discriminate between candidates in a set of calibrated models. As an example, Barde (2016) also compares the Franke and Westerhoff (2012) model to variants of the AR-GARCH model using the MIC. In this prior investigation, however, the former model is not rigorously estimated due to limitations in ABM estimation techniques at the time. Our exercise, by additionally incorporating a rigorous estimation procedure, may thus be viewed as an extension of this previously conducted work.

3.4 Practical Considerations

3.4.1 Choosing the Lag Length

As previously stated, we set $L = 3$ in all estimation experiments involving our proposed method. Naturally, one may wonder whether this is an arbitrary choice or if there is a systematic way of choosing L. Similarly, one may also wonder if the obtained results are robust to this choice, even if only to some extent. We now address both issues.

When applying the proposed methodology, we observed a phenomenon that appeared to be relatively consistent throughout the experiments. In more detail, we observe that while increasing L initially has a pronounced effect on the estimated conditional densities, there exists some $L^* \geq 0$ such that for $L \geq L^*$,

$$p\left(\boldsymbol{x}_{t,i}^{sim} \middle| \boldsymbol{x}_{t-L,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim} : \boldsymbol{\theta}\right) \simeq p\left(\boldsymbol{x}_{t,i}^{sim} \middle| \boldsymbol{x}_{t-L-1,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim} : \boldsymbol{\theta}\right), \tag{3.23}$$

or, in other words, the MDN essentially ignores the additional lags.

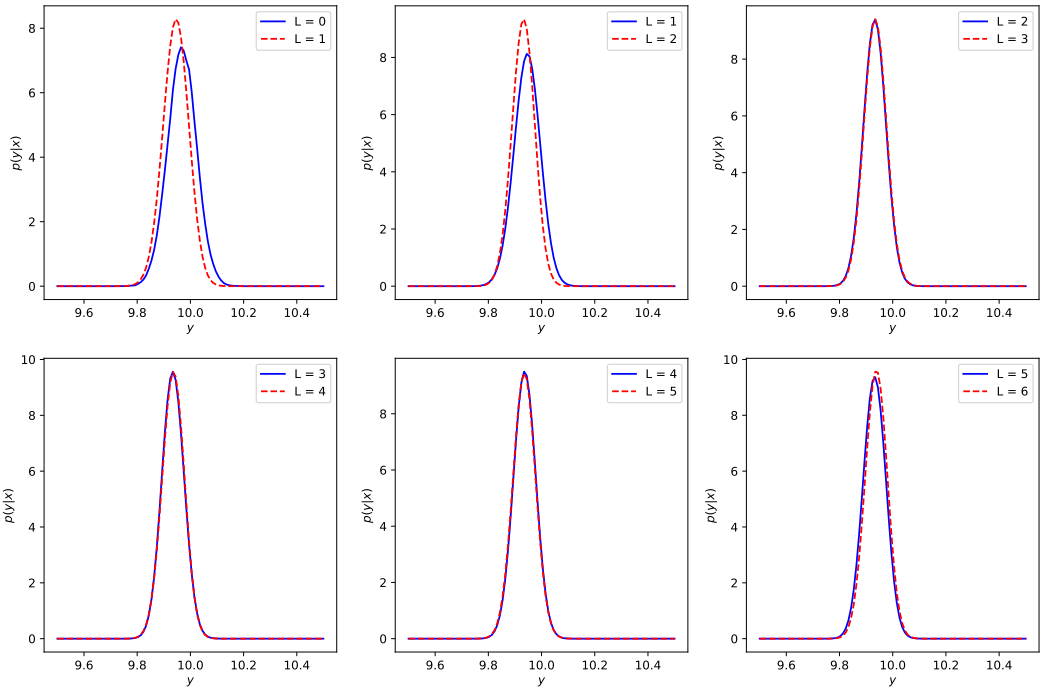


Figure 3.3: A demonstration of the sensitivity of the conditional density estimates to the choice of lag length for a typical example of the Brock and Hommes (1998) model.

We illustrate this graphically in Figure 3.3. Here, we train an MDN using a similar process to that employed in our estimation experiments on an ensemble of 100 realisations of length 1000 generated using the Brock and Hommes (1998) model initialised using the true values associated with the first free parameter set (see Table 3.1). We then randomly draw an arbitrary sequence of 6 consecutive values, $\{z_1, z_2, z_3, z_4, z_5, z_6\}$, from any time series of length 1000 generated by the Brock and Hommes (1998) model for the above parameter

configuration, i.e. any series in the ensemble. This then allows us to use the trained MDN to plot the conditional density functions, $p\left(y\middle|z_{6-L+1},\ldots,z_6\right)$, for differing choices of $L \leq 6$, and observe the aforementioned trend.

In more detail, the upper left panel of Figure 3.3 contrasts the density function obtained when conditioning on none of the previous observations to that obtained when conditioning on a single past realised value. Here we see that, in this case, there is a clear difference in the obtained density functions, with differences in the associated means being apparent and the lag 1 density being significantly more peaked around its mean. A similar trend is also observed in the upper middle panel, which analogously compares the density functions obtained by conditioning on one and two of the previous observations. When examining the additional comparisons represented in subsequent panels, however, we see that the consideration of three as opposed to two previous observations, four as opposed to three previous observations, and so on, results in almost no perceivable difference between the obtained density functions, suggesting that we eventually reach a point, in this case $L^* \simeq 2$, at which the consideration of additional lags has a negligible effect on the obtained conditional density estimates.

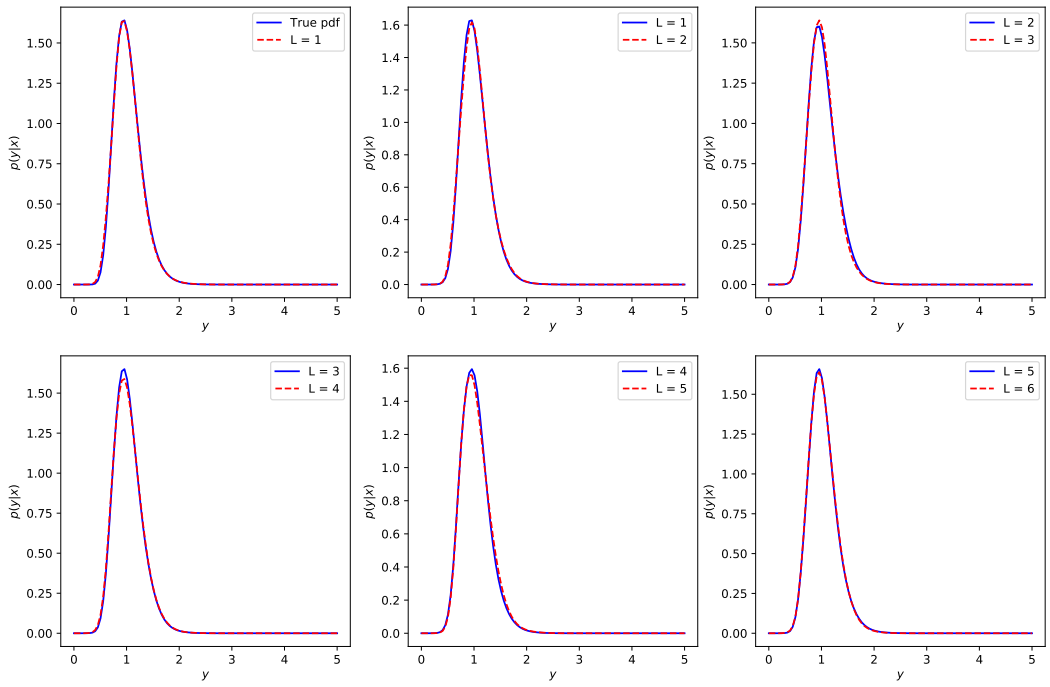


Figure 3.4: A demonstration of the sensitivity of the conditional density estimates to the choice of lag length for i.i.d. random samples following a log-normal distribution, $LN(0, 0.0625)$. The upper left panel compares the conditional density estimate for $L = 1$ to the known, true unconditional density function.

Repeating this exercise on models for which the true number of lags, L_{true}, is known a-priori, we see that $L^* = L_{true}$, providing some evidence that the MDN is able to ignore uninformative lags. To elaborate, Figure 3.4 shows that, when applying the above procedure

to i.i.d. log-normal random samples, the obtained density functions are virtually identical, regardless of the considered number of lags. This is exactly as we would expect, as each observation is, by construction, independent ($L_{true} = 0$), meaning past observations should not have any influence on the resultant density estimates. Analogously, Figure 3.5, which considers a standard AR(2) model, illustrates behaviours consistent with those observed in Figure 3.3, with marked changes in the obtained density functions initially being observed as the number of lags is increased, until reaching $L^* \simeq 2$, at which point the addition of further lags no longer produces a noticeable effect. As in the case of the log-normal samples, this is exactly as we would expect, since the AR(2) model produces, by construction, time series in which observations are dependent only on the last two values ($L_{true} = 2$).

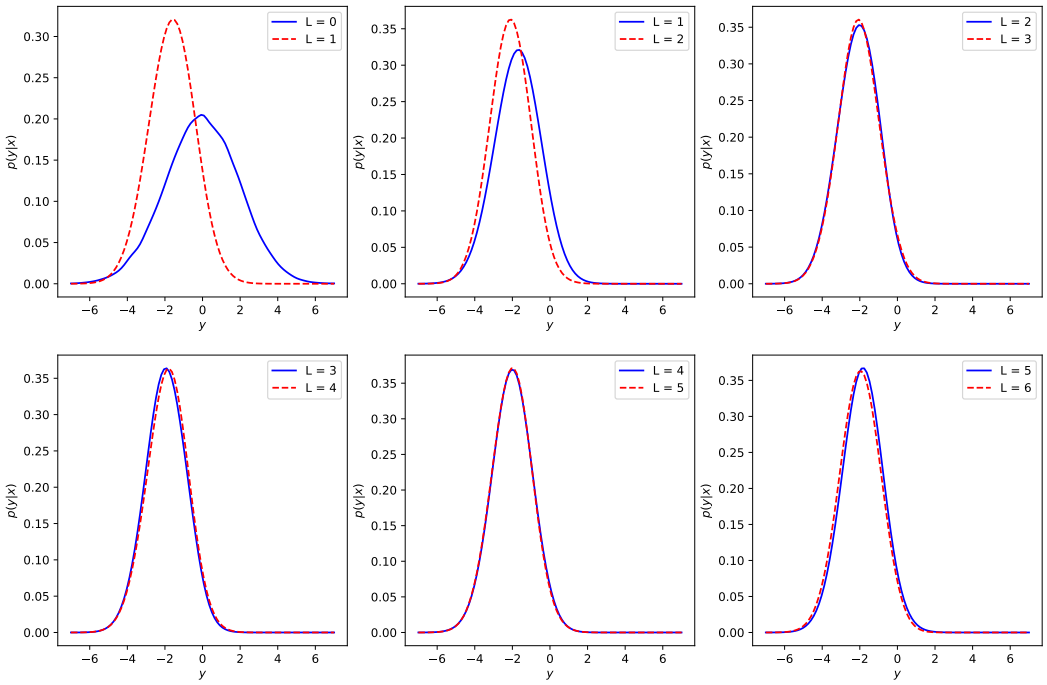


Figure 3.5: A demonstration of the sensitivity of the conditional density estimates to the choice of lag length for an AR(2) model, $x_{t+1} = 0.45x_t + 0.45x_{t-1} + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, 1)$.

This has a number of important implications. Firstly, it implies that plots of the type we have constructed here could be used as a means to systematically inform the choice of L for arbitrary models. Secondly, and perhaps more importantly, it implies that if $L \geq L_{true}$, the procedure should demonstrate at least a modest degree of robustness to the choice of lag at the level of individual conditional density estimates, which would, by extension, result in some level of robustness in the likelihood approximation, provided that the MDN is sufficiently expressive and sufficiently well-trained[26]. This may explain why simply setting

[26]Of course, we are by no means suggesting that this will necessarily hold in all cases, especially in those for which the lag is severely misspecified, since negligible differences at the level of individual observations might accumulate across entire time series, but the above exercises do provide reassuring evidence that there is some scope for misspecification in the chosen number of lags. In particular, we suggest that the interested reader refer to Appendix 3.C.1, where we explore these issues further.

$L = 3$ resulted in a high level of estimation performance in our experiments, regardless of the considered model, since the models considered are not characterised by long-range dependencies.

3.4.2 Computational Costs and Applicability to Large-scale Models

At this point, one may ask whether the proposed estimation routine compares favourably to other contemporary alternatives in terms of computational costs and, perhaps more importantly, whether it may be applied to more complex models. As stated by Grazzini et al. (2017), the cost of generating simulated data using a candidate model is generally dominant, particularly for large-scale models that may need to be run for several minutes in order to generate a single realisation. It is therefore imperative that any estimation methodology keep the simulated ensemble size, which we call R, to a minimum.

As previously stated, we consider at most $R = 100$ replications, which results in a relatively large training set of $R(T_{sim} - L) = 99700$ training examples. This compares favourably to most alternatives in the literature on a number of grounds. Firstly, most studies which have attempted to estimate models of similar complexity make use of ensembles consisting of a far greater number of realisations, typically between 500 and 2000 (Barde 2017; Lamperti 2018a; Lux 2018). Secondly, the training set associated with $R = 100$ is already large relative to the complexity of the network architecture we employ, which is described in Appendix 3.A.4.

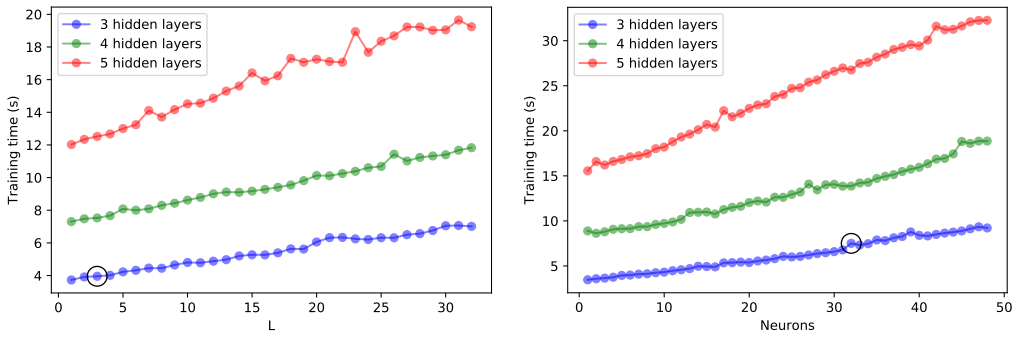


Figure 3.6: Training time for various MDN configurations on the same ensemble of 100 realisations of length 1000 generated using the Brock and Hommes (1998) model initialised using parameter set 1. The point indicated on both the left and right panels corresponds to the configuration employed in our estimation experiments.

To illustrate this point, we repeat the experiment associated with parameter set 1 of the Brock and Hommes (1998) model, changing only the simulated ensemble size, which has been halved to $R = 50$. We find that even with this drastic decrease in the number of Monte Carlo replications, the proposed methodology still performs well and results in a lower loss function value than was obtained using the method of Grazzini et al. (2017) in the original experiments, with a ratio[27] of $LS_{MDN}/LS_{KDE} = 0.2899$. This provides some evidence that

[27]Here LS_{KDE} is determined from the results of the original experiment involving the method of Grazzini et al. (2017) with $R = 100$, while LS_{MDN} is determined from the results of the supplementary experiment involving our proposed methodology with $R = 50$.

even for greatly reduced ensemble sizes, our approach remains viable, and implies that the complexity of the candidate model and hence the employed neural network would likely need to be increased substantially before any increase in R beyond 100 might be required.

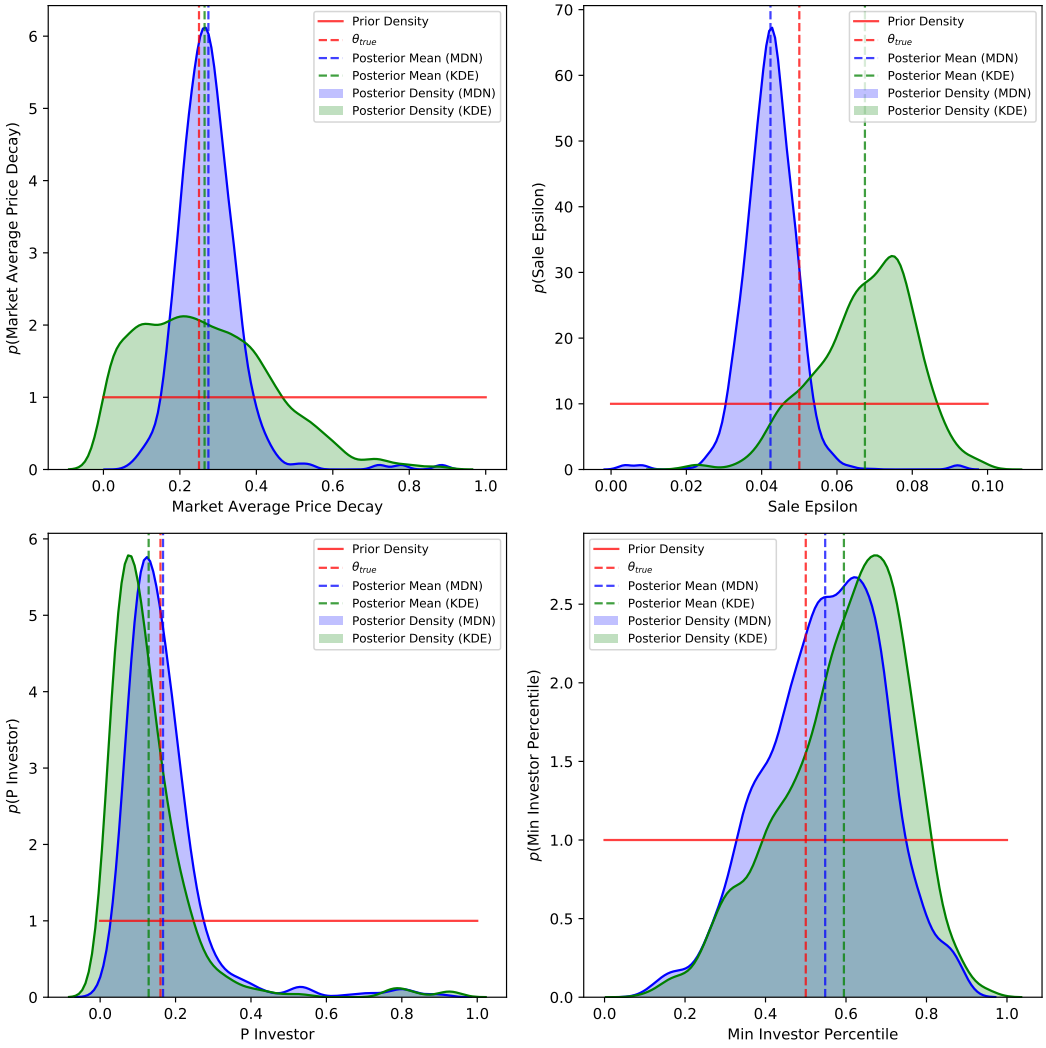


Figure 3.7: Marginal posterior distributions for the housing market model. The plotted density functions are obtained by applying KDE to the posterior samples obtained using each method. The results associated with the method of Grazzini et al. (2017) were originally presented in Section 2.2.2.1.

In addition to concerns related to the size of the simulated ensemble, it is also worthwhile to consider the actual computational costs of the neural network training procedure relative to those associated with the generation of a single model realisation. For this reason, Figure 3.6 demonstrates the training time required by various neural network configurations, most of which are larger than that of the network employed in this investigation, which typically

takes ~ 5 seconds to be trained. We find that even for substantially more complex neural networks than those considered in our investigation, the training time is still typically less than 40 seconds, which compares favourably to the simulation time of large-scale models, and we additionally find that the increase in computational time is linear for both increases in the lag length and network width.

Further, it should be noted that GPU parallelisation was not employed when generating the aforementioned computational cost diagrams. Given the significant speedup that could be expected with the use of such hardware, typically in the region of $20\times$ (Oh and Jung 2004), we find there to be at least some evidence that the time taken to train the neural network should be negligible in comparison to the time taken to generate a single model realisation, even for far more sophisticated neural networks and candidate models.

Of course, a natural next step would be an examination of the ability of our methodology, and particularly the neural network configuration introduced in Appendix 3.A.4, to estimate large-scale models. We therefore repeat the exercise involving the INET Oxford housing market model presented in Section 2.2.2.1. When doing so, we make no changes to the experimental procedure, with the exception of the approach used to estimate the likelihood, where we now consider our proposed methodology, as it is applied in Section 3.3 of this chapter. The results of the above exercise, along with those obtained when instead employing the method of Grazzini et al. (2017), are presented in Figure 3.7, and demonstrate that the performance of our proposed methodology is comparable to that yielded in the case of the simpler models considered in Section 3.3.1. To be more precise, we see that our approach typically results in marginal posteriors that are more strongly peaked around their respective means, with those means tending to, in most cases, be noticeably closer to the true parameter values when compared to those associated with the method of Grazzini et al. (2017). This is particularly evident in the case of the *Sale Epsilon* parameter. Further, we also observe that the aforementioned posterior means are all reasonable estimates of the true parameter values.

It thus follows that the above exercise provides promising evidence that our approach may indeed be suitable for the estimation of large-scale models and further provides a concrete demonstration of the computational tractability of such exercises. This is particularly relevant, as approaches similar to the particle filter considered by Lux (2018) would generally not be well-suited to such models due to their incorporation of high-dimensional latent spaces[28]. Additionally, the probabilistic programming approach employed by Bertschinger and Mozzhorin (2020) cannot be applied to models with discrete agents due to theoretical restrictions induced by the sampling methodology.

Conclusion

In the preceding sections, we have introduced a neural network-based protocol for the Bayesian estimation of economic simulation models (with a particular focus on ABMs) and demonstrated its estimation capabilities relative to a leading method in the existing literature.

[28]Particle filters explicitly sample the space of latent variables, a limiting factor when considering more complex models. As an example, Lux (2018) requires 2000 particles (Monte Carlo replications) to perform a Bayesian estimation exercise involving a simple ABM with a single latent variable. It is therefore reasonable to expect that when dealing with thousands of latent variables, computational feasibility would become a significant concern.

Overall, we find that our methodology delivers compelling performance in a number of scenarios, including the estimation of heterogeneous agent models typically used to test estimation procedures and simple empirical model selection exercises, as well as less orthodox examples, such as identifying dynamic shifts in data generated by a random walk model. In all of the cases tested, we find that our proposed methodology produces estimates closer to known ground-truth values than the approach proposed by Grazzini et al. (2017) and find that it also typically results in narrower and more sharply-peaked posteriors.

In addition to our primary findings, we also discuss practical issues related to the applicability of the proposed routine. We demonstrate that the lag length, which can be viewed as our approach's primary hyperparameter, can be systematically chosen and that the overall estimation performance (relative to the benchmark) demonstrates at least some robustness to this choice. Further, we provide a number of arguments as to the protocol's computational efficiency relative to a number of prominent alternatives in the literature and provide an example application to a state-of-the-art, large-scale model.

3.A Technical Details of the Proposed Estimation Procedure

While we presented an overview of our estimation procedure in Section 3.1, the associated discussions were primarily illustrative and omitted several key details. We thus provide a more technical, step-by-step discussion of our approach in this section.

3.A.1 Training Set Construction

The primary aim of our methodology is the construction of an approximation to the likelihood function for a given set of parameter values, $\boldsymbol{\theta}$. In order to facilitate this process, we make the simplifying assumption that $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})$ depends only on $\boldsymbol{x}_{t-L,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{t-1,i}^{sim}(\boldsymbol{\theta})$, for all $L < t \leq T$. Our problem therefore reduces to the estimation of conditional densities of the form $p\left(\boldsymbol{x}_{t,i}^{sim} \middle| \boldsymbol{x}_{t-L,i}^{sim}, \ldots, \boldsymbol{x}_{t-1,i}^{sim} : \boldsymbol{\theta}\right)$.

In order to estimate the above conditional densities, we will require an appropriate dataset, which is constructed in a number of stages. The first of these stages involves the use of the candidate model to generate an ensemble of R Monte Carlo replications[29], $\boldsymbol{X}^{sim}(\boldsymbol{\theta}, T_{sim}, i), i = i_0, i_0+1, \ldots, i_0+R-1$, for a given value of $\boldsymbol{\theta}$. This is then followed by the construction of two ordered sets for each Monte Carlo replication i in the ensemble,

$$\begin{aligned}\boldsymbol{X}_i^{train}(\boldsymbol{\theta}) = \Big\{ &\left\{\boldsymbol{x}_{1,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{L,i}^{sim}(\boldsymbol{\theta})\right\}, \left\{\boldsymbol{x}_{2,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{L+1,i}^{sim}(\boldsymbol{\theta})\right\}, \ldots, \\ &\left\{\boldsymbol{x}_{T-L,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{T-1,i}^{sim}(\boldsymbol{\theta})\right\} \Big\},\end{aligned} \tag{3.24}$$

and

$$\boldsymbol{Y}_i^{train}(\boldsymbol{\theta}) = \left\{\boldsymbol{x}_{L+1,i}^{sim}(\boldsymbol{\theta}), \boldsymbol{x}_{L+2,i}^{sim}(\boldsymbol{\theta}), \ldots, \boldsymbol{x}_{T,i}^{sim}(\boldsymbol{\theta})\right\}. \tag{3.25}$$

Finally, the sets $\boldsymbol{X}_i^{train}(\boldsymbol{\theta}), i = i_0, i_0+1, \ldots, i_0+R-1$ are concatenated, in order, to produce a single, larger ordered set, $\boldsymbol{X}^{train}(\boldsymbol{\theta})$, with an analogous procedure being applied to $\boldsymbol{Y}_i^{train}(\boldsymbol{\theta})$ to yield $\boldsymbol{Y}^{train}(\boldsymbol{\theta})$.

In essence, $\boldsymbol{X}^{train}(\boldsymbol{\theta})$ consists of rolling windows of length L drawn from the ensemble of Monte Carlo replications, while $\boldsymbol{Y}^{train}(\boldsymbol{\theta})$ consists of the $\boldsymbol{x}_{t,i}^{sim}(\boldsymbol{\theta})$ values that directly follow each window in $\boldsymbol{X}^{train}(\boldsymbol{\theta})$. Together, they form a training set of size $R(T_{sim} - L)$ that can be used to approximate the required conditional densities.

3.A.2 Neural Network Specification and Training

With an appropriate dataset now constructed, we proceed with a more detailed discussion of the MDN itself.

As a starting point, let H be a feedforward neural network with input layer $\boldsymbol{h}_0$ (taking in windows of length L), hidden layers $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_{n-1}$, output layer $\boldsymbol{h}_n$, and weights and biases $\boldsymbol{\psi}$. The mixture parameters are then defined as

$$\boldsymbol{\alpha} = softmax(\boldsymbol{W}_\alpha \boldsymbol{h}_n + \boldsymbol{b}_\alpha), \tag{3.26}$$

[29]The set of R random seeds associated with the R Monte Carlo replications, $i_0, i_0+1, \ldots, i_0+R-1$, while arbitrary, must be kept identical when evaluating the posterior density at different values of $\boldsymbol{\theta}$. As stated by Franke (2009), this is an important nuance that ensures that it is the effect of $\boldsymbol{\theta}$ and not differing sets of random seeds that accounts for differences in posterior density or objective function evaluations. Further, it should also be noted that the same set of random seeds is used for experiments involving the approach of Grazzini et al. (2017) to ensure that the comparative exercises are kept as fair as possible.

$$\boldsymbol{\mu}_k = \boldsymbol{W}_{\mu_k}\boldsymbol{h}_n + \boldsymbol{b}_{\mu_k}, \tag{3.27}$$

and

$$\boldsymbol{\Sigma}_k = diag(\boldsymbol{\sigma}_k^2), \tag{3.28}$$

where $diag(\boldsymbol{x})$ is a diagonal matrix with diagonal $\boldsymbol{x}$ and

$$\log \boldsymbol{\sigma}_k^2 = \boldsymbol{W}_{\sigma_k}\boldsymbol{h}_n + \boldsymbol{b}_{\sigma_k}. \tag{3.29}$$

This results in an expanded neural network with weights and biases

$$\boldsymbol{\phi} = \{\boldsymbol{\psi}, \boldsymbol{W}_\alpha, \boldsymbol{b}_\alpha, \boldsymbol{W}_{\mu_k}, \boldsymbol{b}_{\mu_k}, \boldsymbol{W}_{\sigma_k}, \boldsymbol{b}_{\sigma_k}\} \tag{3.30}$$

that takes windows of length L as input and outputs $\boldsymbol{\alpha}$, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ as defined above.

At this stage, there are a number of nuances worth highlighting. In Eqn. 3.26, notice that we make use of the $softmax$ function[30]. This ensures that the mixture weights, $\boldsymbol{\alpha}$, are strictly positive and sum to one, as required. Additionally, notice that in Eqn. 3.28 we consider a diagonal rather than a full covariance matrix[31]. If we had not made such an assumption, we would have to ensure that the covariance matrices returned by our neural network were positive definite. Though possible in principle, this would significantly increase the number of network parameters and have a potentially detrimental effect on computational performance (Rothfuss et al. 2019). Finally, it should be apparent from Eqn. 3.29 that the neural network outputs a vector of log variances rather than the diagonal covariance matrix, allowing us to avoid imposing positivity constraints on the network output.

Now, all that remains is the training of our constructed network, which is achieved through the application of maximum likelihood estimation to our training set. Denoting by $\boldsymbol{X}_m^{train}$ the m-th entry in $\boldsymbol{X}^{train}(\boldsymbol{\theta})$ (with $\boldsymbol{Y}_m^{train}$ being similarly defined), maximum likelihood estimation is equivalent to solving

$$\arg\min_{\boldsymbol{\phi}} - \sum_{m=1}^{R(T-L)} \log \sum_{k=1}^{K} \alpha_k\left(\boldsymbol{X}_m^{train}\right) \mathcal{N}\left(\boldsymbol{Y}_m^{train} \middle| \boldsymbol{\mu}_k\left(\boldsymbol{X}_m^{train}\right), \boldsymbol{\Sigma}_k\left(\boldsymbol{X}_m^{train}\right)\right) \tag{3.31}$$

using stochastic gradient descent methods, where $\alpha_k\left(\boldsymbol{X}_m^{train}\right)$ is the k-th element of $\boldsymbol{\alpha}\left(\boldsymbol{X}_m^{train}\right)$.

3.A.3 Data Normalisation and Regularisation

While the scheme we have just described could be applied as is, it is likely to perform suboptimally in its current form. This is because neural networks, like most machine learning techniques with a large number of free parameters, have a tendency to overfit the training data and thus perform poorly out-of-sample, particularly when the training set is small (Murphy 2012). In practice, this is often addressed using early stopping, a technique that requires a percentage of the data to be kept separate from the training set in order to

[30] We also do not make use of the subscript k. This is because each mixture component is associated with only a single mixture weight and $\boldsymbol{\alpha}$ refers to the full vector of K weights outputted by the *softmax* function. In contrast, however, $\boldsymbol{\mu}_k$ and $\log \boldsymbol{\sigma}_k^2$ instead refer to the mean and log variance vectors associated with individual mixture components and would thus each have size equal to the number of dimensions (variables) of the simulated data.

[31] It should be noted that the universal density approximation properties of Gaussian mixtures still apply for diagonal covariance matrices.

evaluate out-of-sample performance during each epoch (Prechelt 1998). Such a solution is, however, undesirable in our context, since it requires the generation of additional data, an expensive undertaking for large-scale simulation models.

Fortunately, Rothfuss et al. (2019) present a set of best practices for conditional density estimation using neural networks that provides an alternative solution for overfitting. In particular, a technique called noise regularisation is employed, in which small random perturbations are applied to the data during the training process. It can be shown that this ultimately results in a complexity penalty that favours smoother density estimates that are less prone to overfitting (Rothfuss et al. 2019). For this reason, we apply Gaussian perturbations to training examples in $\boldsymbol{X}^{train}(\boldsymbol{\theta})$ and $\boldsymbol{Y}^{train}(\boldsymbol{\theta})$, which we denote by

$$\boldsymbol{\xi}_x \sim \mathcal{N}(0, \eta_x^2 \boldsymbol{I}) \text{ and } \boldsymbol{\xi}_y \sim \mathcal{N}(0, \eta_y^2 \boldsymbol{I}), \tag{3.32}$$

respectively.

It should be apparent that the degree of regularisation depends directly on the magnitudes of the standard deviations η_x and η_y relative to the range of variation in the training data[32]. This implies that η_x and η_y would have to be adjusted for each candidate model in order to result in the same degree of regularisation. Rothfuss et al. (2019) therefore propose a data normalisation scheme that ensures the training data exhibits zero mean and unit variance, eliminating the need to retune these hyperparameters for each candidate model. This is achieved through the application of a simple transformation to each training example.

Letting $\hat{\boldsymbol{\mu}}_x$ and $\hat{\boldsymbol{\sigma}}_x$ be vectors that contain estimates of the mean and standard deviation along each dimension for training examples in $\boldsymbol{X}^{train}(\boldsymbol{\theta})$, this transformation is given by

$$\tilde{\boldsymbol{X}}_m^{train} = diag(\hat{\boldsymbol{\sigma}}_x)^{-1}(\boldsymbol{X}_m^{train} - \hat{\boldsymbol{\mu}}_x), \tag{3.33}$$

with $\hat{\boldsymbol{\mu}}_y$, $\hat{\boldsymbol{\sigma}}_y$ and $\tilde{\boldsymbol{Y}}_m^{train}$ being defined analogously.

Once the network has been trained on the normalised dataset, we are required to evaluate $\tilde{f}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\phi})$, originally defined in Eqn. 3.8. This is achieved through a simple procedure. Firstly, the normalisation transform is applied to $\boldsymbol{x}$ and $\boldsymbol{y}$ using the same $\hat{\boldsymbol{\mu}}_y$, $\hat{\boldsymbol{\sigma}}_y$, $\hat{\boldsymbol{\mu}}_x$ and $\hat{\boldsymbol{\sigma}}_x$ values used in Eqn. 3.33, yielding $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$. $\tilde{\boldsymbol{x}}$ is then fed through the trained neural network to yield corresponding mixture parameters, allowing us to evaluate the density at $\tilde{\boldsymbol{y}}$, which we denote by $\tilde{g}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\phi}})$. It should be noted that $\tilde{g}$ does not directly correspond to $\tilde{f}$, since we have made a change of variables and the volume of the probability density is not preserved under the normalisation transform for $\hat{\boldsymbol{\sigma}}_y \neq 1$. Rothfuss et al. (2019) do, however, prove that

$$\tilde{f}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\phi}) = \frac{1}{\prod_{j=1}^{J} \hat{\sigma}_y^{(j)}} \tilde{g}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\phi}}), \tag{3.34}$$

where $\hat{\sigma}_y^{(j)}$ is the j-th element of $\hat{\boldsymbol{\sigma}}_y$, allowing us to easily calculate the required density.

3.A.4 Neural Network Architecture

In essence, we have defined a general neural network-based approach to simulation model estimation that is independent of the specific network architecture (number of hidden layers, number of neurons, type of activation functions, and so on) used. Nevertheless, for the sake

[32] As an example, setting $\eta_x = 0.5$ would result in a substantial amount of regularisation for training examples that take values in $[0, 1]$, while essentially having no effect for training examples taking values in $[0, 1000]$.

of completeness, we briefly introduce the (relatively simple) architecture employed in our study, which is used consistently throughout unless stated otherwise.

For the mixture model itself, we set the number of mixture components to be $K = 16$, with the associated mixture parameter network consisting of 3 hidden layers, each with 32 neurons and ReLU activations. This was trained using the well-known Adam optimiser (Kingma and Ba 2015) over 12 epochs[33], with a batch size of 512 and noise regularisation parameters $\eta_x = \eta_y = 0.2$.

The above architecture, which performed well for all of the estimation tasks conducted, was, perhaps rather surprisingly, the first architecture we considered and was chosen by hand rather than through an automated optimisation procedure. Attempts to improve performance by increasing the number of hidden layers, neurons, and mixture components seemed to have little effect (see Appendix 3.C.2), suggesting that the proposed network is sufficiently expressive to produce high-quality density estimates for our considered set of problems.

Further, it should also be noted that, as shown in Section 3.4.2, there is at least some evidence that the above architecture is also effective when confronted with large-scale, complex economic ABMs. We suspect that this will likely hold for other models of similar complexity and therefore make the recommendation that our proposed architecture be used as a baseline for future investigations employing this estimation methodology. Of course, it may certainly be the case that it becomes necessary to construct more expressive networks for some problems and that some form of hyperparameter optimisation would need be carried out. This is beyond the scope of our investigation, however, and we thus leave it to future research.

Finally, it is worth mentioning that neural networks are, as a result of their weight initialisation process and training via stochastic gradient descent, random in nature, meaning that a different set of weights and biases would be learnt each time the same network architecture is trained on the same dataset. It thus follows that the obtained conditional density estimates would, to some extent, be dependent on the random seed, j^*, used to initialise the random number generators associated with the neural network weight initialisation and training procedure. As stated in Footnote 29, it is important that any random seeds be kept consistent across evaluations of the posterior for different values of $\boldsymbol{\theta}$, and this is no exception. Therefore, in line with this discussion, we fix j^* to an arbitrary and non-optimised value, which is kept identical throughout the entire investigation (across all models, parameter sets, and posterior density evaluations). Further discussions and experiments considering the effects of alternative j^* choices are presented in Appendix 3.C.3.

3.B Supplementary Brock and Hommes (1998) Model Exercises

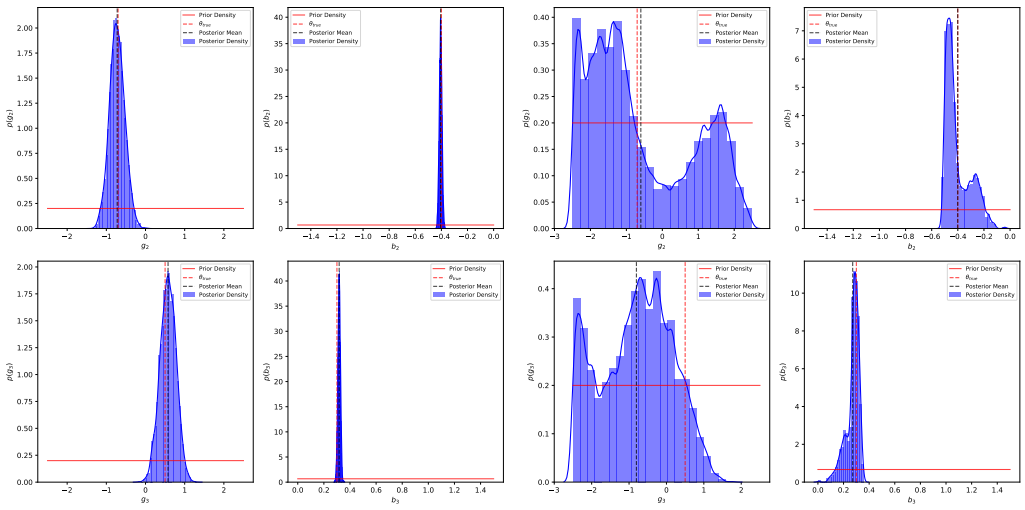
In Section 3.3.1.1, we provided a detailed discussion regarding how restrictions on the ranges of values taken by the trend following and bias parameters associated with each strategy in the Brock and Hommes (1998) model are an intrinsic part of the model definition and further elaborated on how such restrictions are essential for ensuring model identifiability. Along these lines, we assumed $g_2 \in [-2.5, 0]$, $b_2 \in [-1.5, 0]$, $g_3 \in [0, 2.5]$, and $b_3 \in [0, 1.5]$ for the first free parameter set, in line with our definition of a model including a contrarian strategy with a negative bias and a trend following strategy with a positive bias.

[33] Any improvements in the likelihood for subsequent epochs were generally negligible.

We note, however, that the model should, at least in theory, remain identifiable if we were to instead assume $g_2, g_3 \in [-2.5, 2.5]$, since the remaining restrictions on b_2 and b_3 would ensure that the estimation problem remains well-specified[34]. It would thus be interesting to observe if such a change induces any meaningful effects on the performance of both our proposed methodology and that of Grazzini et al. (2017), leading us to repeat the exercises associated with the first free parameter set of the Brock and Hommes (1998) model with these new parameter ranges[35].

Referring to Figure 3.8, which presents a graphical illustration of the results of the above exercise, we observe that the performance of our proposed methodology is unaffected, with it yielding near-identical estimation performance to that observed in the original experiments. The same cannot be said for the method of Grazzini et al. (2017), however, which now produces a rather poorly-behaved bimodal posterior, which can be taken as evidence of the inability of the approach to effectively discriminate between the effects associated with positive and negative values of g_2 and g_3. Further, it should also be noted that although we do provide the posterior means for reference, Murphy (2012) suggests that point estimates are generally uninformative when derived from multimodal posteriors and should thus be interpreted with caution, further highlighting the shortcomings of the approach of Grazzini et al. (2017) in this more challenging context and reinforcing the findings of the original comparative exercise in Section 3.3.1.1.

3.C Robustness Tests



(a) MDN (b) KDE

Figure 3.8: Marginal posterior distributions obtained for free parameter set 1 of the Brock and Hommes (1998) model after relaxing assumptions on the ranges of g_2 and g_3.

[34]Clearly, further relaxation of the parameter range restrictions would result in an estimation problem that is no longer well-specified, as illustrated in Section 3.3.1.1.

[35]Assuming $g_2, g_3 \in [-2.5, 2.5]$ led to very similar overall findings for the second free parameter set.

3.C.1 Number of Lags

In Section 3.4.1, we provided evidence that the conditional density estimates produced by the MDN, and by extension our proposed likelihood approximation, demonstrate some robustness relative to the choice of the number of lags, L. Here, we provide a more complete examination by repeating all of the estimation experiments conducted in Section 3.3.1 that are associated with our methodology, changing only the number of lags, which we have increased to $L = 4$. Referring to the summary presented in Table 3.10, we find that the overall performance of the procedure relative to our chosen benchmark is virtually unchanged[36]. This indicates that any differences resulting from the consideration of the additional lag do not alter the relative rankings of the methods in virtually all cases and thus verifies the robustness of our overall conclusions regarding the competitiveness of our proposed methodology.

Table 3.10: Estimation Result Summary Across All Models for $L = 4$

Outcome	Percentage of Cases
$LS_{mdn} < LS_{kde}$	100
$\lvert\mu^i_{mdn} - \theta^i_{true}\rvert < \lvert\mu^i_{kde} - \theta^i_{true}\rvert$	70.37
$\sigma^i_{mdn} < \sigma^i_{kde}$	85.19

3.C.2 Neural Network Architecture

In all of the preceding experiments involving the application of our proposed methodology, we have employed the neural network architecture described in Appendix 3.A.4 exclusively. Naturally, some readers may wonder if modifying the aforementioned architecture would have a strong effect on the overall estimation performance. For this reason, we aim to provide some insight into the effects of various MDN components on the loss function values through a set of supplementary experiments.

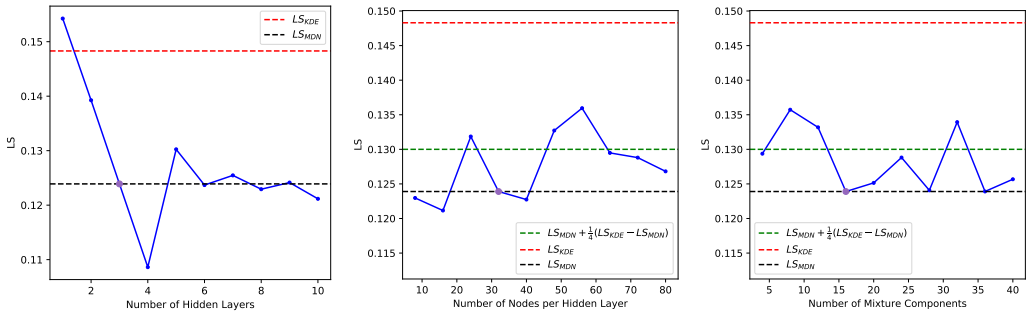
In these experiments, we repeat the exercises associated with the HPM parameter set of the Franke and Westerhoff (2012) model[37], each time modifying a single feature of the MDN (number of hidden layers, number of nodes per layer, and number of mixture components), while keeping the remainder of the network architecture identical to that described in Appendix 3.A.4 and recording the final loss function value.

Our focus on the HPM parameter set of the Franke and Westerhoff (2012) model is justified on a number of grounds. Firstly, in order to obtain a sufficiently large number of data points to deduce loss function trends, we consider 30 different network architectures, each of which is associated with a full Bayesian estimation exercise. This is, of course, a computationally-demanding undertaking that requires us to limit ourselves to a single example case. Additionally, it should also be noted that the performance of the method of Grazzini et al. (2017) is relatively close to that of our proposed methodology in this

[36]Since there are a total of 27 individual parameter cases, the percentage shift in $\sigma^i_{mdn} < \sigma^i_{kde}$ represents a change in only a single binary relation, which happens to be in favour of our proposed methodology.

[37]Deviating slightly from the original experiments, we consider a single chain generated by our chosen adaptive Metropolis-Hastings algorithm in each exercise, primarily due to computational constraints. Note, however, that, as previously discussed, the posterior mean estimates are robust across different chains, making the consideration of a single chain sufficient.

case, with only a very small difference in the loss function values being observed in the original experiments in Table 3.5. It therefore follows that the differences between the loss function values obtained in these previously conducted experiments would provide a challenging benchmark against which to compare the differences induced by the consideration of alternative network architectures.



(a) Network depth (b) Network width (c) Mixture components

Figure 3.9: Loss function values associated with the application of our proposed methodology to the HPM parameter set of the Franke and Westerhoff (2012) model for various neural network architectures. LS_{kde} and LS_{mdn} refer to the loss function values associated with the method of Grazzini et al. (2017) and the default network architecture introduced in Appendix 3.A.4 respectively. The point indicated in purple represents our default architecture.

Elaborating on the above point, we would expect to see one of two behaviours as the complexity of the network is increased. These include a decreasing loss function trend, indicating that the increased network complexity facilitates greater accuracy in the obtained conditional density estimates, or oscillatory behaviour, indicating that the MDN is already sufficiently expressive and that increasing network complexity does not improve performance. In the latter case, these fluctuations emerge due to the stochastic nature of neural networks and the resultant changes in the training process induced by altering the network architecture. Here, it is especially important that the loss function differences induced by these fluctuations are insignificant relative to the performance differences between the methodologies if the overall procedure is to be considered robust. This clearly justifies our choice of the HPM parameter set of the Franke and Westerhoff (2012) model, for which the performance difference is extremely small and for which both methods perform relatively well.

Now, referring to Figure 3.9a, which illustrates the effect of increasing the number of hidden layers, we observe that the loss function value initially exceeds that yielded by the method of Grazzini et al. (2017) and follows a decreasing trend, before reaching a minimum at 4 hidden layers and settling into a series of small oscillations about the loss function value obtained in the original experiments. This initial decreasing trend is not at all surprising, as the number of hidden layers in a neural network has a strong effect on its overall expressiveness and the consideration of too few hidden layers would have a detrimental effect on its ability to approximate functions. The aforementioned convergent behaviour is also exactly as expected in our discussion above. It therefore follows that although it should be ensured that the neural network is sufficiently deep, the consideration of a network with significantly more than 3 or 4 hidden layers is unlikely to lead to meaningful improvements in performance. Further, it appears that the overall estimation performance is relatively stable

for deeper networks, implying that there does not appear to be a significant performance penalty for considering unnecessarily complex networks, other than increased computational time.

Proceeding to Figure 3.9b, which illustrates the effect of increasing the number of nodes per hidden layer, we now observe a somewhat different trend. To summarise, we encounter oscillatory behaviour throughout, with the aforementioned loss function value fluctuations being more or less negligible relative to the differences in method performance. To elaborate, all of the considered network configurations result in loss function values noticeably less than the corresponding value obtained when employing the method of Grazzini et al. (2017) and these loss function values are, in 70% of cases, bounded above by $LS_{mdn} + \frac{1}{4}(LS_{kde} - LS_{mdn})$. This suggests that, in most cases, changes to the network width do not alter the loss function value by more than 25% of the performance difference between the methods. This is particularly relevant, given that the aforementioned performance difference is, as previously discussed, very small. Further, all loss function values not bounded above by this line do not greatly exceed it.

The above factors suggest that increases in the network width are unlikely to lead to improvements in estimation performance and further imply that the choice of the number of nodes per hidden layer is not critical. Interestingly, this is also not surprising. Recent years have seen the emergence of several studies suggesting that network depth is a more important determinant of expressiveness than width (Eldan and Shamir 2016; Lu et al. 2017; Safran and Shamir 2017) and this ultimately seems to have been reflected in our findings. Further, Lu et al. (2017) demonstrate that for ReLU networks, which we consider in our investigation, universal approximation theorems hold for widths of $n + 4$ or greater, where n is the number of inputs. Since we consider exactly 3 inputs (for $L = 3$), all of the network widths presented in Figure 3.9b would meet this criterion and therefore be predicted to be sufficient, likely explaining the lack of improvement in the obtained loss function values for increased widths.

A similar discussion applies to Figure 3.9c, which illustrates the effect of increasing the number of mixture components. Again, we observe fluctuations throughout that are negligible relative to the differences in method performance, and, as before, can offer an explanation for the encountered trends. In more detail, Bertschinger and Mozzhorin (2020) demonstrate that the conditional distribution of returns produced by the Franke and Westerhoff (2012) model can be approximated by a normal distribution. Therefore, even a single mixture component would be sufficient, indicating that the choice of the number of mixture components is not critical and, perhaps more importantly, that the consideration of significantly more mixture components than is required, such as the 16 we employ in our investigation, does not result in a significant performance penalty, as in the case of the network depth and width.

Based on the examples given above, it thus follows that there is at least some evidence that our approach may be robust to the choice of neural network architecture and number of mixture components, provided that the number of hidden layers is sufficient, and that increasing the complexity of the considered network, while not detrimental, does not seem to result in much improvement in estimation performance. We therefore suggest that, in future, if the default architecture presented in Appendix 3.A.4 does not seem sufficient for more complex problems, that emphasis should be placed on increasing the number of hidden layers, rather than the number of mixture components or nodes per layer.

3.C.3 Neural Network Random Seed

Recall that in Appendix 3.A.4, we indicated that the random seed associated with the neural network weight initialisation and training procedures is fixed to an arbitrary value, j^*, and kept identical for every posterior density evaluation performed in this investigation, in other words, across all $\boldsymbol{\theta}$ values and models. As previously explained, this is necessary to ensure that it is the effect of changes in the parameter values and not randomness in the initial network weights that underlies differences between subsequent posterior density evaluations.

Table 3.11: Neural Network Seed Robustness Testing for the Franke and Westerhoff (2012) Model

	α_0	α_n	α_p	σ_c
Param Set HPM				
$\boldsymbol{\theta}_{true}$	-0.327	1.79	18.43	2.087
MDN				
$\boldsymbol{\mu}_{posterior}$	-0.1800	1.9230	17.0413	2.2707
	(0.0034)	(0.0016)	(0.0310)	(0.0082)
$\boldsymbol{\sigma}_{posterior}$	0.0808	0.0667	2.3295	0.2767
	(0.0021)	(0.0014)	(0.0285)	(0.0042)
LS		**0.1278 (0.0013)**		

All values represent the means over 50 neural network seeds with the associated standard errors in brackets.

Since our initial choice of j^* is entirely arbitrary, the same value is used throughout this investigation for many different models and parameter sets, and our proposed methodology performs relatively well in all of these cases, it would appear that either initial weight configurations that perform well in one example are likely to perform well in others[38], or alternatively, that for many problems, the choice of j^* may not be particularly important. Naturally, a definitive answer to the above would be extremely difficult to determine. Nevertheless, it is likely that some readers would, as in the case of the underlying neural network architecture, wish to be given some indication of the sensitivity of some of our results to alternative choices of j^*.

Therefore, much like in the case of our experiments involving various neural network architectures, we repeat the exercises involving the application of our proposed methodology to the HPM parameter set of the Franke and Westerhoff (2012) model a total of 50 times[39], each time considering a different value for j^*. This ultimately results in a set of 50 estimates for the posterior means of the parameters, posterior standard deviations of the parameters, and the final loss function value that may be used to determine various descriptive statistics.

As in Appendix 3.C.2, the use of a single example model[40] is necessitated due to computational costs. Further, the consideration of the HPM parameter set of the Franke and

[38]There is now a fairly extensive literature that clearly demonstrates the effects of a poor initial weight configuration on final performance (see Glorot and Bengio (2010), Mishkin and Matas (2016), and Gaier and Ha (2019) for examples).

[39]As in the case of the experiments involving different network architectures in Appendix 3.C.2, we consider a single Markov chain per seed.

[40]This is, of course, a relevant caveat. The determination of the extent to which these and the findings in

Westerhoff (2012) model is again justified due to the relatively small difference in performance between the method of Grazzini et al. (2017) and our proposed methodology in this context.

To elaborate, we would generally expect that, if our results were robust to the choice of j^*, loss function differences arising from different seed choices would be small relative to the performance difference between the methods, or, in other words, that choosing a different random seed would not lead to our methodology being beaten by the benchmark in most cases. Clearly, in such an exercise, the obtained results would be most informative when the performance difference between the methods is small.

Now, referring to Table 3.11, which presents the means and standard errors[41] of various quantities of interest over the 50 considered neural network random seeds, we observe some evidence of robustness for the considered example. Firstly, notice that the averages of the posterior means, posterior standard deviations, and the final loss function value over the ensemble demonstrate a high degree of similarity when compared to the equivalent values obtained for our single fixed random seed in Table 3.5. Additionally, notice that, in all cases, the associated standard errors are also relatively small.

Appendix 3.C.2 hold for other examples and in general is ultimately beyond the scope of this investigation due to computational constraints, as stated above.

[41] The standard error is given by $\frac{\tilde{\sigma}}{\sqrt{N}}$, where $\tilde{\sigma}$ is the sample standard deviation over the neural network random seeds and N is the total number of seeds considered.

Chapter 4

Bayesian Model Selection for Inter- and Intraday Financial Agent-based Models

In the preceding chapters, we have focused exclusively on the problem of model estimation, both when comparing existing methods and when proposing our own. Though extremely important, estimation remains only one of a number of key tasks associated with statistical inference. An equally-important undertaking is that of model selection, or alternatively, the determination of which among a set of estimated candidate models best describes the real-world data-generating process. Since alternative model specifications may lead to different predictions or policy outcomes, it is vital that sensible and well-motivated choices be made when determining which model to employ for the task at hand.

In the ABM context, this is particularly crucial. Firstly, as alluded to in Chapter 1, model selection exercises could be used to demonstrate the superiority of ABMs relative to traditional approaches and provide more compelling evidence for the adoption of the paradigm, which still remains controversial in some circles. Secondly, there now exist many competing ABM frameworks attempting to model a wide variety of financial and economic phenomena, with many of these models now differing substantially due to the flexibility offered by adopting a simulation-based paradigm. Model selection exercises could thus play an important role in refining this broad literature and determining which frameworks are best to develop further in future work.

In line with the above, we now demonstrate how the estimation method developed in Chapter 3 may be extended to perform rigorous and widely-applicable Bayesian model selection and exhaustively test the proposed techniques on two broad classes of models, inter- and intraday financial ABMs, which we introduced and contrasted in more detail in Chapter 1. In the interday case, which has also been considered in prior literature (Franke and Westerhoff 2012; Barde 2016; Bertschinger and Mozzhorin 2020), we demonstrate the ability of our proposed protocol to replicate a number of existing key findings. In the intraday case, which, to the best of our knowledge, has not yet been considered, we demonstrate how the protocol may be used to compare the two major competing paradigms for order book modelling: the behavioural and ZIA approaches.

4.1 Bayesian Predictive Model Selection

In order to effectively compare a set of models, it is necessary to first quantify some notion of model quality. While one could consider several different notions in such a context, perhaps the most influential paradigm is the assessment of the ability of a candidate model to make reasonable out-of-sample predictions. A model which is incapable of making good predictions is unlikely to be a reasonable representation of the system of interest and, in much the same way, a well-specified model is more likely to make predictions that are at least somewhat reasonable. In line with the above, the use of the predictive density, or equivalently the likelihood of a fitted model evaluated on out-of-sample data, has been widely-applied in differing ways throughout the general model selection literature[1].

To be more precise, we consider the function

$$p(\tilde{\boldsymbol{X}}|\boldsymbol{\theta}, \boldsymbol{X}), \tag{4.1}$$

where $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \ldots, \tilde{\boldsymbol{x}}_{\tilde{T}}\}$ is out-of-sample data, $\boldsymbol{\theta}$ is the set of model parameters, $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ is in-sample data used to estimate $\boldsymbol{\theta}$, and $\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t \in \mathbb{R}^n$ for all t.

In a Bayesian context, estimation results in a distribution for $\boldsymbol{\theta}$ rather than a point estimate, leading us to consider the average over the posterior,

$$\begin{aligned} \log p_{post}(\tilde{\boldsymbol{X}}) &= \log \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})}\left[p(\tilde{\boldsymbol{X}}|\boldsymbol{\theta})\right] \\ &= \log \int p(\tilde{\boldsymbol{X}}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{X}) d\boldsymbol{\theta}. \end{aligned} \tag{4.2}$$

It should be apparent that the above quantity depends on the specific instance of out-of-sample data and the probability thereof under the true model, necessitating the computation of the expectation

$$\mathbb{E}_f\left[\log p_{post}(\tilde{\boldsymbol{X}})\right] = \int \log p_{post}(\tilde{\boldsymbol{X}}) f(\tilde{\boldsymbol{X}}) d\tilde{\boldsymbol{X}}, \tag{4.3}$$

where f represents the density function associated with the true model.

While the above is the ideal metric by which to compare competing models, it cannot be computed, as f is unknown. This necessitates the use of approximations. In practice, some form of cross-validation would generally be the first choice to provide a reasonable and somewhat robust approximation to $\mathbb{E}_f\left[\log p_{post}(\tilde{\boldsymbol{X}})\right]$. Unfortunately, this would require us to fit a candidate model many times and would thus be far too computationally-demanding in our case, since ABM estimation, even if performed only once, is notoriously challenging and requires large quantities of simulated data to be produced. This ultimately necessitates the consideration of more practical alternatives.

4.1.1 The Deviance Information Criterion

As an alternative to cross-validation, one might instead consider evaluating Eqn. 4.2 on the in-sample data, $\boldsymbol{X}$, as opposed to the out-of-sample data, $\tilde{\boldsymbol{X}}$, eliminating the need to consider the expectation over f. This resulting approximation, though easy to compute, will, in general, be prone to overestimation and, additionally, favour more complex and flexible models that are likely to overfit the in-sample data and have poor generalisation performance. This necessitates the addition of some form of complexity penalty.

[1] Our presentation here is largely based on discussions included in the review by Gelman et al. (2014).

Perhaps the most famous of such constructions is the celebrated Akaike Information Criterion[2] (AIC), given by

$$\text{AIC} = -2\log p(\boldsymbol{X}|\hat{\boldsymbol{\theta}}) + 2k, \tag{4.4}$$

where k is the number of model parameters (complexity penalty) and $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate obtained by fitting the model to $\boldsymbol{X}$.

The above construction, while influential and widely-used in practice, nonetheless suffers from key weaknesses. In particular, while employing the number of model parameters as a measure of model complexity is suitable in certain contexts, such as linear models with uniform priors, it may not necessarily be appropriate for more complex models. As an example, one may find that the addition of certain parameters may have limited effects on model flexibility, while the addition of others may be far more significant. Further, the considered parameters may interact with one another in complex ways, non-uniform priors may induce regularising effects (effectively reducing model complexity), and complex latent variable and hierarchical structures may further complicate the defensibility of a complexity penalty determined only by a simple parameter count.

In this, the need for an easily-calculable and data-dependent complexity penalty is clear. While there have historically been many attempts at such a construction, the Deviance Information Criterion (DIC), originally introduced by Spiegelhalter et al. (2002) almost two decades ago, has remained influential and been successfully applied throughout the Bayesian inference literature, but is, perhaps surprisingly, yet to be comprehensively applied in the context of financial ABMs, leading us to consider it here.

In short, the DIC is given by

$$\text{DIC} = \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})}\left[D(\boldsymbol{\theta})\right] + p_{\text{DIC}}, \tag{4.5}$$

where $D(\boldsymbol{\theta}) = -2\log p(\boldsymbol{X}|\boldsymbol{\theta})$ is called the deviance and

$$p_{\text{DIC}} = \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})}\left[D(\boldsymbol{\theta})\right] - D(\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})}\left[\boldsymbol{\theta}\right]). \tag{4.6}$$

The term p_{DIC}, introduced above, is a data-dependent complexity penalty constructed[3] such that it reduces to k (asymptotically as T becomes large) for linear models with uniform priors, in line with what one would expect in contexts where the AIC is known to perform well. For this reason, p_{DIC} is sometimes referred to as the effective number of parameters and the DIC can be seen as a Bayesian generalisation of the AIC[4]. In other contexts, however, where a simple parameter count would no longer be appropriate, p_{DIC} overcomes the limitations of the AIC and behaves intuitively. To illustrate this point, linear models regularised through the use of non-uniform priors yield p_{DIC} values less than k, in line with the reduced model complexity promoted by the priors. Further, in more complex cases, such as stochastic volatility models for financial returns, Berg et al. (2004) and Chan and Grant (2016) demonstrate that p_{DIC} might also exceed k due to the presence of latent variables that promote increased flexibility without increasing the number of model parameters.

While the above construction may indeed seem promising at face value, the DIC has also been subject to criticism, as summarised (and in some cases rebutted) by Spiegelhalter

[2]Note that the criterion is derived from a frequentist perspective and involves the multiplication of $\log p(\boldsymbol{X}|\hat{\boldsymbol{\theta}})$ by -2. This multiplication has become standard in the literature and results in a quantity referred to as the deviance.

[3]See Spiegelhalter et al. (2002) for a more detailed discussion of the associated derivation.

[4]The DIC can also be equivalently expressed as $-2\log p(\boldsymbol{X}|\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})}\left[\boldsymbol{\theta}\right]) + 2p_{DIC}$, making this connection even more apparent.

et al. (2014) in a follow-up to the original DIC paper. Much of the aforementioned criticism is philosophical in nature and deals with issues that are, even today, somewhat contentious within the model selection community. As an example, Spiegelhalter et al. (2014) are generally sceptical of the notion of the existence of a true model and argue that such a notion is, in general, a mathematical convenience that tends to be of little value in practical circumstances. This is ultimately reflected in their discussion of the DIC, which is more concerned with the accuracy of short-term predictions than the identification of the true model[5]. A detailed treatment of this underlying debate is, of course, beyond the scope of this investigation, which simply aims to determine whether the DIC is a useful model selection tool in the context of ABMs. We therefore instead focus on more practical criticisms of the DIC that would directly impact its performance in our context.

To be more precise, the most relevant criticisms of the DIC are the fact that p_{DIC} is not invariant to reparameterisation, such that innocuous transformations of the parameter space, such as taking logs, may result in slightly different DIC values, and, additionally, the fact that p_{DIC} may sometimes be negative when the posterior mode and mean are not approximately equal[6]. In response to these criticisms, Gelman et al. (2004) propose an alternative formulation,

$$p_{\text{DIC-V}} = \frac{1}{2} Var_{p(\boldsymbol{\theta}|\boldsymbol{X})} \left[D(\boldsymbol{\theta}) \right], \tag{4.7}$$

that ensures that the complexity penalty is always positive (even if the posterior mean and mode are very different), while still maintaining the desirable properties of p_{DIC} discussed previously.

4.1.2 The Watanabe-Akaike Information Criterion

As previously stated, the DIC has, despite some criticism, remained a core component of the Bayesian model selection toolbox for almost two decades, having been widely-applied in a variety of contexts. Interestingly enough, Spiegelhalter et al. (2014) highlight that although a number of alternative model selection criteria have been proposed in subsequent years, the vast majority of these have failed to catch on and remain comparatively obscure. The most prominent of these alternatives, and the only one to have received significant attention, is the Watanabe-Akaike Information Criterion (WAIC) (Watanabe 2010).

Although not explicitly stated, it should be clear that the DIC is expressed in terms of the likelihood over the entire set of observations, $p(\boldsymbol{X}|\boldsymbol{\theta})$. In contrast to this, the WAIC is instead defined in terms of the individual observation-level likelihoods, $p(\boldsymbol{x}_t|\boldsymbol{\theta})$, leading to a model selection criterion given by

$$\text{WAIC} = -2 \sum_{t=1}^{T} \log \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})} \left[p(\boldsymbol{x}_t|\boldsymbol{\theta}) \right] + 2 p_{\text{WAIC}}, \tag{4.8}$$

with complexity penalty

$$p_{\text{WAIC}} = 2 \sum_{t=1}^{T} \log \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})} \left[p(\boldsymbol{x}_t|\boldsymbol{\theta}) \right] - 2 \sum_{t=1}^{T} \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})} \left[\log p(\boldsymbol{x}_t|\boldsymbol{\theta}) \right]. \tag{4.9}$$

[5]Nevertheless, and as previously stated, models that produce more accurate predictions are more likely to also be well-specified and therefore structurally similar to the unknown data-generating process, even if such a mapping is imperfect. We demonstrate this explicitly in Section 4.4.1.1, where it is shown that the DIC can indeed correctly identify the model that produced a given set of simulated data for a variety of examples.

[6]In other words, the posterior deviates significantly from a normal distribution.

While the above does share some cosmetic similarities with the DIC, the theoretical underpinnings of the WAIC are vastly different and the WAIC is generally considered to have greater theoretical support when compared to the more heuristic DIC[7]. Nevertheless, this increased theoretical justification has not led to the widespread adoption of the WAIC in practice and the DIC remains comparatively popular.

This is mostly due to the fact that the WAIC can be quite difficult to implement efficiently and robustly (Spiegelhalter et al. 2014) due to its requirement that the likelihood be factorised into observation-level components, which can lead to computational and conceptual difficulties, particularly for time series data[8]. This is in contrast to the computation of the DIC, which involves the calculation of a number of simple point estimates and is therefore generally trivial. Additionally, it should also be noted that, subsequent to the initial presentation of the WAIC by Watanabe (2010), more compelling and user-friendly alternatives also making use of observation-level likelihoods, such as Pareto smoothed importance sampling leave-one-out (PSIS-LOO) cross-validation (Vehtari et al. 2017), were introduced, and quickly supplanted the WAIC.

4.1.3 Pareto Smoothed Importance Sampling Leave-one-out Cross-validation

The aforementioned model selection approaches, namely the AIC, DIC, and WAIC, are all instantiations of the same underlying principle: evaluating predictive accuracy on in-sample data and correcting for model complexity in order to account for the possibility of overfitting. Although this paradigm may seem quite distinct, Gelman et al. (2014) point out that such methodologies do, in fact, share conceptual links to cross-validation and can be shown to be asymptotically equivalent to leave-one-out cross-validation (LOO-CV) for certain models and under certain assumptions. Of course, this equivalence does not apply to all models and the associated assumptions do not always hold in practice. It therefore follows that the estimates of out-of-sample predictive accuracy obtained using LOO-CV and one of the aforementioned criteria may not necessarily be equivalent.

This naturally leads to an alternative approach: attempting to directly approximate the LOO-CV estimate of out-of-sample predictive accuracy in such a way that the overall computational burden is reduced. We now discuss how this may be achieved in practice.

We begin by assuming that the candidate model has a factorisable likelihood of the form

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{\theta}), \tag{4.10}$$

or, in other words, a likelihood that is expressable as the product of individual observation-level contributions[9]. The associated LOO-CV estimate of out-of-sample predictive accuracy is then given by

$$\text{LOO-CV} = \sum_{t=1}^{T} \log p(\boldsymbol{x}_t|\boldsymbol{x}_{-t}), \tag{4.11}$$

[7]Unfortunately, a detailed treatment of the theoretical underpinnings of the WAIC is far beyond the scope of this investigation, though the interested reader should refer to Spiegelhalter et al. (2014) and Gelman et al. (2014) for more detailed discussions.

[8]The DIC, on the other hand, has been shown to be easy to apply in the time series context, particularly for stochastic volatility models (Berg et al. 2004; Chan and Grant 2016).

[9]We make use of observation-level contributions of the form $p(\boldsymbol{x}_t|\boldsymbol{\theta})$ for notational convenience, but $p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{\theta})$ and $p(\boldsymbol{x}_t|\boldsymbol{x}_{-t}, \boldsymbol{\theta})$, where $\boldsymbol{z}_t$ is a set of latent variables, are also admissible (Vehtari et al. 2019).

where

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{-t}) = \int p(\boldsymbol{x}_t|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{x}_{-t})d\boldsymbol{\theta}, \tag{4.12}$$

and $\boldsymbol{x}_{-t}$ refers to all observations in $\boldsymbol{X}$ excluding $\boldsymbol{x}_t$.

In essence, we are simply fitting the model to every observation in $\boldsymbol{X}$, excluding $\boldsymbol{x}_t$, to obtain a posterior distribution over the model parameters, $p(\boldsymbol{\theta}|\boldsymbol{x}_{-t})$, that is then used to determine the out-of-sample predictive accuracy for $\boldsymbol{x}_t$. This process is then repeated for every observation in $\boldsymbol{X}$, resulting in a set of T likelihood-based, out-of-sample predictive accuracy estimates that we ultimately use to approximate $\mathbb{E}_f\left[\log p_{post}(\tilde{\boldsymbol{X}})\right]$. Unfortunately, and as previously alluded to, this is prohibitively expensive from a computational perspective, as it requires us to estimate the model parameters T times.

Fortunately, a possible solution to the above problem does exist and relies on an important observation originally made by Gelfand et al. (1992). Specifically, they note that it is possible to construct the approximation

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{-t}) \approx \frac{\sum_{s=1}^{S} r_t^s p(\boldsymbol{x}_t|\boldsymbol{\theta}^s)}{\sum_{s=1}^{S} r_t^s}, \tag{4.13}$$

where $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^S$ are samples from $p(\boldsymbol{\theta}|\boldsymbol{X})$, since

$$r_t^s = \frac{1}{p(\boldsymbol{x}_t|\boldsymbol{\theta}^s)} \propto \frac{p(\boldsymbol{\theta}^s|\boldsymbol{x}_{-t})}{p(\boldsymbol{\theta}^s|\boldsymbol{X})}. \tag{4.14}$$

Therefore, by fitting the model to $\boldsymbol{X}$ in its entirety only once and reweighting the observation-level likelihood contributions in Eqn. 4.12 by the importance ratios r_t^s defined in Eqn. 4.14, we are able to obtain estimates of the out-of-sample predictive accuracy for each data point in $\boldsymbol{X}$, without the need to refit the model. This ultimately results in a far less computationally-demanding approximate cross-validation method commonly referred to as importance sampling leave-one-out (IS-LOO) cross-validation.

While IS-LOO cross-validation does indeed seem promising at face value, it is seldom used in practice. This is because, as pointed out by Vehtari et al. (2017), the importance ratios often have large or infinite variance, which can severely impact the accuracy of the approximation. In response, Vehtari et al. (2017) propose a smoothing procedure that involves fitting a generalised Pareto distribution to the 20% largest importance ratios, resulting in a more numerically-stable version of the IS-LOO approximation referred to as PSIS-LOO cross-validation.

It should be noted, however, that although the above smoothing procedure always results in a set of modified importance ratios that have finite variance, this does not necessarily imply that the resultant estimate of $\log p(\boldsymbol{x}_t|\boldsymbol{x}_{-t})$ is accurate. Fortunately, as a byproduct of the above smoothing procedure, a diagnostic for determining the reliability of the final approximation is also obtained. In more detail, the shape parameter, $\hat{k}_t$, of the fitted generalised Pareto distribution may be used to determine the existence of the moments of the unmodified importance ratio distribution, which, in turn, may be used to determine whether the central limit theorem or variations thereof apply. This ultimately indicates whether the accuracy of the estimate improves as the number of posterior samples is increased. Vehtari et al. (2017) consider three main scenarios[10]:

[10] A $\hat{k}_t$ value is produced for each observation $\boldsymbol{x}_t$ in $\boldsymbol{X}$ and it should generally be ensured that the number of $\hat{k}_t$ values exceeding 0.7 is negligible relative to T and further that none exceed 1.

- $\hat{k}_t < 0.5$, which implies that the variance of the unmodified importance ratios is finite, the central limit theorem holds, and convergence is rapid. Clearly, the estimate is reliable in this case.
- $0.5 \leq \hat{k}_t \leq 1$, which implies that although the variance of the unmodified importance ratios is infinite, the mean does exist and the generalised central limit theorem for stable distributions therefore applies. Convergence in this case is less rapid, however, and Vehtari et al. (2017) suggest that, in their experience, the estimate is generally reliable up to values of $\hat{k}_t$ not greatly exceeding 0.7, which they thus suggest as a cutoff point. While the variance of the modified importance ratios would still be finite and convergence still theoretically achievable for larger values of $\hat{k}_t \leq 1$, the required number of posterior samples may be prohibitively large and the estimate may therefore not be reliable for a finite number of posterior samples.
- $\hat{k}_t > 1$, which implies that the mean and variance of the raw importance ratios do not exist and therefore neither form of the central limit theorem applies. Clearly, the estimate is unreliable in this case.

In light of the above, it should be no surprise that PSIS-LOO cross-validation has come to be an extremely popular tool for Bayesian model selection, since it not only provides a fast and numerically-stable approximation to LOO-CV, but additionally also provides a straightforward method to diagnose whether the final approximation is reliable. The latter is a likely reason for its more widespread adoption than the WAIC, which, although having strong theoretical support, lacks a similar diagnostic[11].

4.1.3.1 Leave-one-out and Leave-future-out Cross-validation

At this point, a number of subtle nuances and potential caveats should be highlighted. First and foremost, while LOO-CV may indeed be applied to time series data[12] (which our investigation focuses on), it could be argued that some variant of leave-future-out cross-validation (LFO-CV) may be more appropriate.

In more detail, LFO-CV begins with the fitting of a candidate model to a window of initial size L consisting of consecutive observations drawn from $\boldsymbol{X}$, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L\}$, yielding a posterior distribution over the parameters, $p(\boldsymbol{\theta}|\boldsymbol{x}_{1:L})$, where $\boldsymbol{x}_{1:L} = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L$. This is then followed by the evaluation of a likelihood-based measure of the out-of-sample predictive accuracy for the next M observations in $\boldsymbol{X}$,

$$p(\boldsymbol{x}_{L+1:L+M}|\boldsymbol{x}_{1:L}) = \int p(\boldsymbol{x}_{L+1:L+M}|\boldsymbol{x}_{1:L}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{x}_{1:L}) d\boldsymbol{\theta}, \tag{4.15}$$

after which $\boldsymbol{x}_{L+1}$ is added to the window and the preceding steps are repeated. After several

[11] Additionally, Vehtari et al. (2017) are also strongly associated with the popular probabilistic programming language *Stan*, which has also contributed to the popularity of PSIS-LOO cross-validation.

[12] An excellent discussion on the use of LOO-CV in the context of time series is provided by Aki Vehtari in a blog post available at `https://statmodeling.stat.columbia.edu/2015/01/16/loo-waic-time-series`. In essence, he argues that, in a time series context, LOO-CV essentially indicates how effectively a model replicates the structure or dynamics of the time series in question and can be used to determine which, among a set of candidate models, most accurately captures the temporal evolution of the considered system. Of course, he does warn that care must be taken when one is actually concerned with forecasting itself rather than prediction as a means for model comparison, though this is not a concern in our investigation given our focus on the latter.

iterations, the procedure ultimately yields a final window of size $T-M$ and the overall LFO-CV estimate of out-of-sample predictive accuracy may then be calculated according to

$$\text{LFO-CV} = \sum_{t=L}^{T-M} \log p(\boldsymbol{x}_{t+1:t+M}|\boldsymbol{x}_{1:t}). \tag{4.16}$$

Clearly, the above is a more natural approach in a time series context, particularly when forecasting is the primary aim of the modelling exercise. Nevertheless, several factors make LFO-CV an impractical choice in our investigation. First and foremost, LFO-CV is, like LOO-CV, extremely computationally-demanding, requiring a candidate model to be estimated a substantial number of times. While these computational constraints can be bypassed in the LOO-CV case using PSIS, making only a single estimation run necessary, this is not quite the case for LFO-CV. To elaborate, Bürkner et al. (2020) apply PSIS to LFO-CV in a recent paper and demonstrate that while the resultant procedure is able to reduce the number of required refits by a factor of roughly 25 to 100, a non-negligible number of refits is still necessary. Indeed, assuming similar levels of reduction also apply in our case, where we primarily consider time series of length $T = 2000$ or greater, a minimum of approximately 20 to 80 refits would be required, unfortunately rendering the procedure computationally-intractable[13].

Of course, we are not suggesting that computational limitations alone are an adequate justification for our consideration of PSIS-LOO cross-validation and it is thus important to highlight additional supporting arguments:

- Firstly, it should be noted that, in addition to their construction of the PSIS-LFO methodology, Bürkner et al. (2020) also identify a fundamental link between the PSIS-LOO and PSIS-LFO estimates of out-of-sample predictive accuracy. In general, PSIS-LOO cross-validation can be seen as an upward-biased estimate of LFO-CV for $M = 1$. When the considered time series data exhibits a clear temporal trend (i.e. linear or quadratic), this bias is typically non-negligible and can be quite large. Crucially, however, Bürkner et al. (2020) demonstrate that this bias is substantially reduced when considering trendless time series data. Therefore, since our investigation focuses exclusively on asset price and index log-returns, which typically fluctuate around a more or less constant mean, such a bias is likely to be less pronounced in our case and therefore less of a concern.

- Secondly, there is already precedent in the literature for the use of PSIS-LOO cross-validation in the context of financial ABMs. In more detail, Bertschinger and Mozzhorin (2020) make use of the probabilistic programming language *Stan* to estimate several small-scale financial ABMs using S&P 500 log-returns and demonstrate that PSIS-LOO cross-validation may be applied successfully to the estimated models to identify the best-performing candidate. This success is ultimately not surprising, since, as stated in our first point above, PSIS-LOO should be appropriate for return data, as considered in their exercise. While there are a number of key similarities between

[13]This is especially true in the case of the LOB models introduced in Section 4.3.2, where a single Bayesian estimation exercise may require several days of computing time to be run to completion, even if the Monte Carlo replications making up the simulated ensemble used to estimate the likelihood for a given value of $\boldsymbol{\theta}$ are generated in parallel. This is because, in virtually all cases, the theoretical minimum time required to evaluate the likelihood is equal to the minimum time required to generate a single Monte Carlo replication of the model. While parallelisation can minimise increases in computational time associated with generating multiple replications, each replication would still require the same amount of time to run to completion.

our investigation and that of Bertschinger and Mozzhorin (2020), there are also some important differences. For example, Bertschinger and Mozzhorin (2020) make use of the *Stan* language to specify the candidate models and estimate the associated likelihoods. While such tools are generally appropriate for small-scale models that aim to approximate a daily time-scale, they are not suitable for more complex ABMs, such as LOB models that aim to approximate intraday time-scales, since such models often consist of thousands of discrete components (individual limit orders), and *Stan* requires models to be expressable in terms of a continuous state space. This investigation may therefore be viewed as an extension of the work of Bertschinger and Mozzhorin (2020) in its application of PSIS-LOO to models of a larger scale for which *Stan* would not be appropriate.

4.2 Experimental Procedure

In the preceding sections, we have focused primarily on introducing and comparing a number of standard tools in the Bayesian predictive model selection literature. We now provide a more concrete discussion regarding how we apply them to discriminate between a number of different financial ABMs in this investigation.

4.2.1 Likelihood Approximation and Bayesian Estimation

At this point, it should be apparent that the aforementioned approaches all depend critically on being able to evaluate the likelihood of a candidate model, whether for the complete set of empirical data, $p(\boldsymbol{X}|\boldsymbol{\theta})$, or at the observation level, $p(\boldsymbol{x}_t|\boldsymbol{\theta})$, and being able to perform some form of Bayesian estimation. Recall, however, that since ABMs typically do not have tractable likelihood functions, this necessitates some form of numerical approximation, which can be nontrivial, especially for more complex ABMs. Fortunately, Chapter 3 provides an ideal tool for use in this context in the form of a neural network-based likelihood approximator, which, when combined with the adaptive Metropolis-Hastings algorithm of Griffin and Walker (2013), results in a widely-applicable Bayesian estimation protocol, which we employ throughout this chapter.

In more detail, we begin by calibrating a candidate model to a set of empirical time series data, $\boldsymbol{X}$, using the Bayesian estimation routine introduced in Chapter 3, yielding parameter samples distributed according to $p(\boldsymbol{\theta}|\boldsymbol{X})$ and corresponding log-likelihood values, $\log p(\boldsymbol{X}|\boldsymbol{\theta})$, which constitute samples from the posterior over the log-likelihood. The obtained posterior log-likelihood samples may then be used to compute the DIC associated with the estimated candidate model according to

$$\mathrm{DIC} \simeq \bar{D}(\boldsymbol{\theta}) + \frac{1}{2(S-1)} \sum_{s=1}^{S} \left[(-2 \log p(\boldsymbol{X}|\boldsymbol{\theta}^s) - \bar{D}(\boldsymbol{\theta}))^2 \right], \tag{4.17}$$

where $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^S$ are samples from $p(\boldsymbol{\theta}|\boldsymbol{X})$ and $\bar{D}(\boldsymbol{\theta}) = \frac{1}{S} \sum_{s=1}^{S} \left[-2 \log p(\boldsymbol{X}|\boldsymbol{\theta}^s) \right]$.

By calculating the DIC for each in a set of candidate models, one obtains directly comparable scores that may be used to rank each model in the set, with the model resulting in the lowest DIC value generally predicted to be the best among the candidates[14].

[14]When determining whether a DIC score difference between two models is significant, i.e. large enough to suggest that one model is indeed likely to yield a higher out-of-sample predictive accuracy than another,

In addition to allowing for the direct calculation of the DIC, the above routine may also be used to determine the PSIS-LOO estimate of out-of-sample predictive accuracy[15]. Recall, from Section 3.1.3, that our considered likelihood approximation approach involves, as a key step, the expression of the likelihood of the empirical data as a product of estimated conditional densities of the form

$$\tilde{f}\left(\boldsymbol{x}_{t-L,i}^{sim},\ldots,\boldsymbol{x}_{t-1,i}^{sim},\boldsymbol{x}_{t,i}^{sim},\boldsymbol{\phi}\right)\simeq p\left(\boldsymbol{x}_{t,i}^{sim}\middle|\boldsymbol{x}_{t-L,i}^{sim},\ldots,\boldsymbol{x}_{t-1,i}^{sim}:\boldsymbol{\theta}\right). \tag{4.18}$$

Clearly, these estimated conditional densities correspond directly to the observation-level likelihood contributions required by PSIS-LOO cross-validation and it therefore follows that they may be stored for each posterior sample[16] during the running of the adaptive Metropolis-Hastings algorithm and eventually used to calculate the PSIS-LOO estimate, as outlined in Eqns. 4.11-4.14.

As with the DIC, we calculate the PSIS-LOO estimate (multiplying by -2 to obtain a distance-based measure in the same vein as the AIC and DIC) for each in a set of candidate models, select the model corresponding to the lowest distance value, and consider PSIS-LOO score differences greater than 5 between competing models to be significant.

4.2.2 Simulation-based Validation

By employing the protocol discussed above in Section 4.2.1 to facilitate the calibration of several models to empirical datasets and hence the calculation of the associated DIC and PSIS-LOO estimates of out-of-sample predictive accuracy, it should be relatively straightforward to devise a comprehensive set of experiments that attempt to determine the most appropriate model specification for different markets, assets and time-scales. While it is not uncommon in the literature to simply apply similar procedures, determine the final criterion values, assume these values to be credible by default, and draw conclusions regarding which model is most appropriate based on these values, we believe that this could lead to flawed insights that should be guarded against. Indeed, as shown by Celeux et al. (2006), there exist examples for which approximations of out-of-sample predictive accuracy such as the DIC, though ubiquitous in the literature, perform poorly and fail to correctly identify the true data-generating process in exercises involving pseudo-empirical data.

A good practice, therefore, is to follow the example set by Celeux et al. (2006) and conduct similar simulation-based validation exercises before proceeding to empirical data[17].

---

we follow the guidance provided on David Spiegelhalter's DIC webpage, `https://www.mrc-bsu.cam.ac.uk/software/bugs/the-bugs-project-dic`, and consider differences greater than 5 to be meaningful.

[15] While we could of course also calculate the WAIC, we have decided against this, both due to the WAIC's relative lack of popularity in practical applications and since Vehtari et al. (2017) find PSIS-LOO cross-validation to deliver superior performance.

[16] Recall, from Section 2.1.3.4, that the adaptive Metropolis-Hastings algorithm introduced by Griffin and Walker (2013) generates a series of S' sample sets, each containing N parameter samples. Since, for our default settings, $N = 70$ and $S' = 15000$, this would correspond to over 1050000 parameter samples, storing the observation-level likelihoods for every sample would be intractable from the perspective of memory and storage space. Therefore, as a more practical alternative, we instead consider the parameter samples obtained at the end of a given estimation experiment and apply a thinning procedure in which a single, randomly-selected sample is drawn from every 5th sample set (following the burning-in period). We then recompute the observation-level likelihoods for each of the samples that constitute this thinned Markov chain. Since each sample set is, as a whole, distributed according to the posterior distribution (after burning-in), this would correspond to drawing 1000 posterior samples of the observation-level likelihoods in the case of a typical burning-in period of 10000 and a total of $S' = 15000$ sample sets.

[17] Berg et al. (2004) and Chan and Grant (2016) conduct similar exercises to demonstrate the ability of the DIC to discriminate between several stochastic volatility model variants.

To be more precise and using notation analogous to that of Chapters 2 and 3, we begin by letting $\boldsymbol{X}_M^{sim}(\boldsymbol{\theta}, T, i)$ be the output of an arbitrary model, M, in a set of candidate models, $\boldsymbol{M} = \{M_1, M_2, \ldots, M_{N_M}\}$, for parameter set $\boldsymbol{\theta}$, output length T, and random seed i. We then consider $\boldsymbol{X} = \boldsymbol{X}_{M_{true}}^{sim}(\boldsymbol{\theta}_{true}, T_{emp}, i^*)$, where M_{true} is a chosen model in $\boldsymbol{M}$, as a proxy for real data to which each model in $\boldsymbol{M}$ may be calibrated and for which the associated DIC and PSIS-LOO values may be calculated. In such a case, we would expect that, if the DIC and PSIS-LOO were effective model selection tools (when underpinned by the Bayesian estimation protocol developed in Chapter 3), that M_{true} would be assigned the best out-of-sample predictive accuracy scores, or, in other words, that the correct data-generating process would be selected.

We conduct a number of validation exercises of the type described above, each time considering a different true model specification and generating pseudo-empirical data of length $T_{emp} = 2000$. When performing the requisite Bayesian estimation of each candidate model, we consider simulated ensembles of $R = 50$ Monte Carlo replications of length $T_{sim} = 2000$ to approximate the likelihood, and iterate the adaptive Metropolis-Hastings algorithm for a total of 15000 iterations (with the first 10000 sample sets discarded as part of a burning-in period). We repeat this process a total of 5 times (unless otherwise stated) to generate a total of 5 Markov chains, each containing parameter and log-likelihood samples from the posterior. A DIC value can be calculated for each of these chains and we return the minimum DIC value obtained across these chains as the final DIC score for the candidate model. We then proceed to calculate the corresponding PSIS-LOO estimate for comparison with the DIC.

As a final remark, it should be apparent that although we generate multiple Markov chains for each model, we consider only a single chain when calculating the final out-of-sample predictive accuracy estimates. This is a step inherited from the work of Bertschinger and Mozzhorin (2020) and is necessary due to complications introduced into the sampling process by certain models. As an example, recall, from Section 3.3.2, that the Franke and Westerhoff (2012) model may, in certain cases, give rise to local posterior modes due to output equivalences existing between certain parameter configurations for which the values of σ_f and σ_c are more or less flipped[18]. In most cases, local modes of this type are at a substantial distance from the global mode and the sampling procedure may therefore, in some situations, become trapped in a suboptimal (i.e. having a much lower mean log-likelihood) area of the parameter space for certain initialisations of the algorithm[19]. Clearly, the consideration of samples from suboptimal areas of the parameter space would result in less accurate DIC and PSIS-LOO estimates and may therefore lead to unfair underestimates of the out-of-sample predictive accuracy for some models, leading us to employ the multi-chain procedure described above[20].

Of course, even if the sampler may sometimes become trapped in a problematic region of the parameter space for a given example (a specific model and empirical dataset combination), this does not imply that this is a ubiquitous phenomenon affecting most examples,

[18]Recall that this arises due to idiosyncrasies associated with a given model's parameters and their effects on the model's output rather than the sampling or likelihood estimation algorithms. Indeed, despite our overall approaches to both aspects differing substantially from that of Bertschinger and Mozzhorin (2020), we observe similar phenomena to those encountered in their investigation.

[19]While this phenomenon appears to be most strongly associated with the Franke and Westerhoff (2012) model, we also encountered similar behaviours for the ARMA-GARCH model

[20]In general, selecting the chain associated with the minimum DIC value corresponds to a balanced tradeoff between maximising the mean and minimising the variance of the in-sample log-likelihood, ensuring that the final distribution results in the best possible fit to the data, while still being numerically-stable.

and even in cases where such behaviour is observed, we find that it usually only affects some of the chains for a given exercise, with the best scoring chains usually having converged to the same distribution. To illustrate this point, we also consider the top 3 chains in each set, as measured by their associated DIC scores, and calculate the Gelman and Rubin (1992) convergence diagnostic using the log-likelihood samples obtained from these chains. In virtually all cases, the obtained diagnostic value suggests that the best chains have converged to the same distribution. This can be taken as evidence that, in most cases, the top 3 chains do, in fact, reflect convergence to the true posterior, even if some of the other chains in the set may have converged to problematic local modes[21].

4.3 Candidate Models

With the employed model selection tools now described in detail, we proceed with a description of the models that constitute our sets of candidates. As previously pointed out, we focus on models that aim to replicate the asset return dynamics associated with two distinct time-scales, specifically inter- and intraday regimes, leading to two sets of models, which we denote by $\boldsymbol{M}_{inter}$ and $\boldsymbol{M}_{intra}$ respectively.

4.3.1 Interday Models

Our first set of candidate models, $\boldsymbol{M}_{inter}$, consists primarily of ABMs that take the form of systems of coupled equations that involve some form of market clearing mechanism that determines a closing price for each trading day by balancing the demands associated with each of the model's trader types. Such models have now become ubiquitous in the ABM estimation and model selection literatures and are often presented as more interpretable alternatives to standard models of daily asset returns and volatility, such as ARMA-GARCH and variations thereof. While initially a fairly niche area of research, recent years have seen increasing interest in practical demonstrations of the ability of ABMs of this class to outperform traditional approaches, with the work of Barde (2016) and Bertschinger and Mozzhorin (2020) being key examples. Our exercises involving these models can therefore be viewed as a verification of prior findings through the use of new methods[22].

4.3.1.1 Revisited Models

Beginning our discussion of the considered models, we note that, in addition to a number of new candidates, we also include a relevant subset of the models first introduced in Chapters 2 and 3. These include the ARMA$(2, 2)$-ARCH(2) and Franke and Westerhoff (2012) models, full descriptions of which are provided in Sections 2.1.2 and 3.2 respectively.

In addition to the previously considered HPM and WP variants of the Franke and Westerhoff (2012) model, we also include a new variant that includes both the wealth and herding (WH) mechanisms and hence defines the attractiveness of fundamentalism relative to chartism at time t as

$$a_t = \alpha_w(w_t^f - w_t^c) + \alpha_n(n_t^f - n_t^c), \tag{4.19}$$

where the definitions of the above symbols are provided in Section 3.2.

[21] In rare cases for which the initial set of chains does not yield an appropriate Gelman and Rubin (1992) convergence diagnostic, additional chains are generated until we obtain agreement with respect to the top 3. This ensures that each model is given a fair assessment.

[22] This replication of prior findings would also act as a validation exercise for our model selection protocol.

4.3.1.2 ARMA$(2,2)$-GARCH$(1,1)$ Model

The first new model we introduce is a variation of the previously considered ARMA$(2,2)$ model, defined according to

$$\begin{aligned} r_{t+1} &= a_1 r_t + a_2 r_{t-1} + b_1 \sigma_t \epsilon_t + b_2 \sigma_{t-1} \epsilon_{t-1} + \sigma_{t+1} \epsilon_{t+1}, \\ \sigma_{t+1}^2 &= \omega + \alpha_1 \epsilon_t^2 + \beta_1 \sigma_t^2, \end{aligned} \tag{4.20}$$

where $\epsilon_t \sim \mathcal{N}(0,1)$.

From the above, it should be clear that we have replaced the ARCH(2) error term with a GARCH$(1,1)$ equivalent. As described in Section 3.2.3, the GARCH framework is a respected and near-ubiquitous approach to the modelling of volatility and is therefore frequently combined with an ARMA foundation to model asset returns. For this reason, the above model acts as the primary benchmark against which we compare the performance of various ABMs in our investigation[23].

4.3.1.3 Alfarano et al. (2008) Model

The second and final new model we discuss is that originally introduced by Alfarano et al. (2008), which, much like its main competitor, the Franke and Westerhoff (2012) model, simulates the interactions between two populations of traders, specifically fundamentalists, who sell overvalued and buy undervalued assets, and noise traders, who buy or sell based on their current sentiment (optimistic or pessimistic). As in the case of many interday financial ABMs, the core elements of the model may be expressed as a system of coupled equations[24]:

$$r_{t+1} = (f_{t+1} - f_t) + (x_{t+1} - x_t), \tag{4.21}$$

$$x_t = \frac{2n_t}{N} - 1, \tag{4.22}$$

$$f_{t+1} - f_t = \sigma_f \epsilon_t, \tag{4.23}$$

where $\epsilon_t \sim \mathcal{N}(0,1)$, r_t is the log-return at time t, f_t is the log fundamental value at time t, x_t is the overall noise trader population sentiment at time t, n_t is the number of noise traders currently having an optimistic outlook at time t, N is the total number of noise traders, and $\sigma_f > 0$ is the standard deviation of the random walk governing the evolution of the fundamental value.

From Eqn. 4.21, it should be apparent that the return dynamics are driven by two distinct components: one associated with fundamentalist traders that essentially exhibits random walk behaviour, and another associated with noise traders that tracks changes in the overall population sentiment, which is positive (negative) when the majority of the noise traders in the simulation have an optimistic (pessimistic) outlook. Therefore, much of the model's complexity is a direct result of changes in sentiment exhibited by individual noise traders throughout the simulation.

[23] In the illustrative example in Section 3.3.2, we considered a slightly less complex AR-GARCH model as a benchmark, which has now been extended by additionally incorporating two moving average lags. This provides a more challenging benchmark, in line with the broadened scope of this investigation.

[24] We consider the model as it is presented in the work of Chen and Lux (2018) and Lux (2018), which the interested reader may wish to refer to for additional details regarding the derivation of the aforementioned system of equations.

These changes in sentiment are rooted in a simple herding mechanism implemented in the form of Poisson processes, where it is assumed that pessimistic agents transition to an optimistic state at rate

$$\pi_t^+ = a + bn_t, \tag{4.24}$$

and optimistic agents transition to a pessimistic state at rate

$$\pi_t^- = a + b(N - n_t), \tag{4.25}$$

where $a > 0$ determines the base transition rate and $b > 0$ determines the strength of the herding component.

In practice, the above Poisson processes are simulated using the well-known Gillespie algorithm. It thus follows that the waiting time to the next event (trader agent switching from an optimistic to a pessimistic state or vice versa) at some time t is drawn from an exponential distribution with parameter

$$\lambda = (N - n_t)\pi_t^+ + n_t\pi_t^- . \tag{4.26}$$

The event type is then determined according to a uniform random draw, with a single optimistic trader agent becoming pessimistic with probability $n_t\pi_t^-/\lambda$, or a single pessimistic trader agent becoming optimistic with probability $(N - n_t)\pi_t^+/\lambda$, after which the transition rates are recalculated and the time of the next event determined[25].

As a final remark, we discuss the initialisation of the model. Following the example of Lux (2018), the initial population sentiment is drawn randomly according to the probability density function

$$p(x) = \frac{\Gamma(2\xi)}{2^{2\xi-1}\Gamma(\xi)^2}(1 - x^2)^{\xi-1}, \tag{4.27}$$

where $\xi = \frac{a}{b}$, after which the initial noise trader population composition and transition rates may be determined.

4.3.2 Intraday Models

With the first set of models now described, we proceed to our second, $\boldsymbol{M}_{intra}$. This set consists primarily of models that simulate the submission of limit and market orders to a central LOB by various trader agents over a series of trading sessions. In contrast to the ABMs contained in $\boldsymbol{M}_{inter}$, there is no direct market clearing mechanism that balances supply and demand; rather, orders are matched directly to one another, resulting in a LOB that evolves through time and defines the current mid-price. As previously discussed in Section 1.2, this is ultimately a far more realistic representation of actual continuous trading and is the primary reason for the applicability of such models at more granular time scales.

Interestingly, models of the above class are seldom considered in the model selection and estimation literatures[26] and our exercises involving them therefore represent, to the best of our knowledge, one of the first applications of modern Bayesian model selection tools to quantitatively compare the two major LOB modelling paradigms: behavioural and ZIA models.

[25] Multiple switch events may occur in a single time step, since the waiting times between events may, of course, be less than 1 for certain combinations of a and b. r_{t+1} would thus be calculated once all the switch events occurring between times t and $t + 1$ have been executed.

[26] See Platt and Gebbie (2018) for an early attempt employing MSM.

4.3.2.1 Smith et al. (2003) Model

We now describe the first and simplest LOB model we consider, that of Smith et al. (2003), which is widely considered to be a canonical example of the ZIA paradigm. In many ways, this model may also be viewed as the framework upon which the more sophisticated Preis et al. (2006) model, which we will discuss shortly, is built. For this reason, we consider a variation of the Smith et al. (2003) model based on discussions by Preis et al. (2007), which makes this connection more explicit.

Like most agent-based LOB modelling approaches, Smith et al. (2003) model simulations proceed in terms of a series of trading windows or time steps[27], $t = 1, 2, \ldots, T$, during which trader agents submit limit or market orders to a central LOB. As time progresses, this results in changes in the composition of orders in the LOB and ultimately the mid-price.

All trading decisions are random in nature and follow Poisson processes, with market and limit orders being placed at fixed rates of μ and α per time step respectively[28], and existing limit orders being cancelled at a rate of δ per order per time step. As in the case of the Alfarano et al. (2008) model, we implement the above using the Gillespie algorithm, where the waiting time to the next trading event at some time t is drawn from an exponential distribution with parameter

$$\lambda = \mu + \alpha + \delta N_t, \tag{4.28}$$

where N_t is the total number of limit orders in the order book at time t. The trading event is a limit order submission with probability $\frac{\alpha}{\lambda}$, a market order submission with probability $\frac{\mu}{\lambda}$, and a limit order cancellation with probability $\frac{\delta N_t}{\lambda}$. Once an event occurs, the current simulation time is updated and the time of the next event determined. Once every event occurring between a given set of time steps, t and $t+1$, has been executed, the mid-price of the order book is recorded and determines the prevailing market price at time $t+1$, which we denote by p_{t+1}.

Regarding the trading events themselves, limit orders to buy or sell are placed with equal probability, and have size drawn randomly from a half-normal distribution with standard deviation[29] $\sqrt{2\pi}$. The associated limit order price is drawn uniformly from the interval

$$[p_a - p_{lim}\Delta, p_a - \Delta], \tag{4.29}$$

in the case of a limit buy order, and

$$[p_b + \Delta, p_b + p_{lim}\Delta], \tag{4.30}$$

in the case of a limit sell order, where p_b and p_a are the current best bid and ask in the order book respectively, and Δ is the tick size. In all cases, the final order price is rounded to the nearest multiple of the tick size.

In the original formulation of the model by Smith et al. (2003) and in contrast to the approach adopted here, limit orders are placed over an infinite price interval. While this

[27] Generally, the interpretation of a single time step or trading window is dependant on the specific application of the model and is likely to depend on the empirical dataset to which the model is fitted. As an example, one might consider each time step or trading window to correspond to a single minute of trading when the target dataset represents a series of one-minute mid-price bars.

[28] In the original model by Smith et al. (2003), α is a rate per unit price and unit time. This is adapted to instead be a rate per unit time by Preis et al. (2007), since the length of the interval over which limit orders are placed is a free parameter in the modified formulation.

[29] Smith et al. (2003) originally consider $\sqrt{2\pi}\sigma$, where σ is a model parameter. Preis et al. (2007), however, indicate that the inclusion of this extra parameter is not essential for the recovery of the original model behaviours and thus do not include it, leading us to do the same.

facilitates the analytical work conducted in the original investigation, fitting the model to return time series data requires the addition of a mechanism that allows the scale of the mid-price volatility to be adjusted. Preis et al. (2007) achieve this through the introduction of the parameter $p_{lim} > 1$, which controls the length of the interval over which limit orders are placed (see Eqns. 4.29 and 4.30). In general, smaller (larger) values of p_{lim} lead to smaller (larger) average mid-price movements per time step.

The remaining trade events are trivial to implement. Order cancelation simply involves the removal of a random order from the LOB. Market orders, which are either buy or sell orders with equal probability, simply reduce the size of the best ask (σ_a) or best bid (σ_b) by the corresponding size of the placed buy or sell order respectively[30]. In cases for which the size of the placed order is greater than σ_a (σ_b), the best ask (bid) is removed from the order book, the size of the current market order is reduced by σ_a (σ_b), and the above process is repeated until the placed market order is fully-executed. As in most real financial markets, orders are always matched according to price-time priority[31].

Naturally, some form of initialisation is required before continuous trading is simulated in order to ensure that there is a sufficient number of existing orders in the LOB. We consider a similar initialisation to that discussed in Preis et al. (2007), where 100 initial limit orders are placed about an initial mid-price of $p_a = p_0 = p_b$.

As a final remark, it should be noted that in the original work by Smith et al. (2003), the primary focus is an investigation into the effects of the model's parameters on the long-run statistical properties of the simulated market, which includes various aspects of the associated microstructure, such as the spread, depth profile, price impact function, and volatility. In contrast to this, our investigation instead focuses on the simulated mid-price time series generated by various candidate LOB models and attempts to quantify the relative similarity of log-return dynamics produced by these models to those observed in equivalent series downsampled from empirical LOB data. This may also be viewed as the determination of how well a given model acts as an intraday mid-price simulator, and thus more closely resembles the aims of the work of Preis et al. (2007).

In line with the above focus on mid-price time series, and much like in the work of Platt and Gebbie (2018), which considers the related Preis et al. (2006) model, we treat the order book as a latent structure of variable dimension that drives the dynamics of the resultant mid-price time series, and therefore consider all 4 model parameters to be free parameters that are adjusted during the application of our Bayesian estimation procedure[32].

4.3.2.2 Preis et al. (2006) Model

Following on from the Smith et al. (2003) model, we now introduce a second LOB modelling approach, that of Preis et al. (2006), which extends the former in a number of meaningful ways.

Much like the model of Smith et al. (2003), simulations progress through a series of time steps, $t = 1, 2, \ldots, T$, with trader agents submitting limit and market orders to a central LOB during each trading window, t to $t+1$, and the mid-price being recorded at the end of each time step. Unlike the model of Smith et al. (2003), however, which does not explicitly define agent types or population sizes, Preis et al. (2006) describe a market consisting of two

[30] As in the case of limit orders, market order sizes are drawn randomly from a half-normal distribution with standard deviation $\sqrt{2\pi}$.

[31] This implies that buy (sell) limit orders with higher (lower) prices always execute first and that in cases where two limit orders have identical prices, the one that was placed earliest is executed first.

[32] This will remain our standard approach for all of the considered LOB models.

groups comprised of N_A traders each, liquidity providers and liquidity takers. Each liquidity provider submits limit orders at a rate of α per time step and cancels their existing limit orders at a rate of δ per order per time step. Analogously, liquidity takers submit market orders at a rate of μ per time step.

In practice, this essentially implies that limit and market orders are submitted to the order book at rates of αN_A and μN_A respectively and that existing limit orders are cancelled at a rate of δN_t. Therefore, despite being formulated in terms of distinct trader populations, the order arrival and cancellation processes in the Preis et al. (2006) model are equivalent to those defined in the Smith et al. (2003) model, with the population size parameter, N_A, essentially acting as a scaling factor for the limit and market order arrival rates[33]. Model initialisation is also virtually identical, with $\lfloor 10\alpha N_A \rfloor$ initial limit orders being placed about an initial mid-price of $p_a = p_0 = p_b$ before the commencement of continuous trading.

Despite the above similarities, however, differences emerge when considering the Preis et al. (2006) order placement mechanisms in more detail:

- Firstly, the probabilities of buy and sell market orders are no longer equal and vary with time[34]. In more detail, the probability of a given market order being a bid to buy at time t is denoted by $q_{taker}(t)$ and follows a mean-reverting random walk with mean $q_{taker}(0) = \frac{1}{2}$, increment Δ_S per unit time, and a mean reversion probability of $\frac{1}{2} + \left| q_{taker}(t) - \frac{1}{2} \right|$. As demonstrated by Platt and Gebbie (2018), this change allows the model to reproduce the non-trivial order flows exhibiting long memory often found in empirical investigations.

- Secondly, limit orders are no longer placed uniformly over a fixed interval and liquidity providers instead employ an adaptive order placement mechanism. To elaborate, a limit order placed at time t has price[35]

$$p_a - 1 - \eta, \tag{4.31}$$

if a buy order, and price

$$p_b + 1 + \eta, \tag{4.32}$$

if a sell order, where η is an exponentially distributed random number, generated according to

$$\eta = \lfloor -\lambda(t) \ln(u) \rfloor \tag{4.33}$$

and $u \sim U(0, 1)$. The placement depth process, $\lambda(t)$, is defined according to

[33] While the Gillespie algorithm would certainly be applicable, we instead adopt a batch-like approach to event resolution that is more in line with the original model description. In more detail, Preis et al. (2007) note that, given the rates described above, approximately αN_A limit orders and μN_A market orders would be placed and δN_t limit orders cancelled during each time step. Therefore, in order to improve computational efficiency, the same sequence of events is carried out, in order, during each time step: the submission of $\lfloor \alpha N_A \rfloor$ limit orders, the submission of $\lfloor \mu N_A \rfloor$ market orders, the removal of each existing order in the order book with probability δ, the measurement of the mid-price, the iteration of $\lambda(t)$, and the iteration of $q_{taker}(t)$. A more detailed description of the overall flow of events is presented in Figure 2 of Platt and Gebbie (2018).

[34] The probability of buy and sell limit orders does, however, remain identical.

[35] Unlike in our description of the Smith et al. (2003) model, where prices are measured in units of currency, all prices are now measured in terms of the number of ticks. Therefore, before calculating the final log-return time series used in the model comparison exercise, we divide each observation of the obtained Preis et al. (2006) model mid-price time series by the number of ticks in a single unit of currency, p_{scale}.

$$\lambda(t) = \lambda_0 \left(1 + \frac{\left| q_{taker}(t) - \frac{1}{2} \right|}{\sqrt{\left\langle \left[q_{taker}(t) - \frac{1}{2} \right]^2 \right\rangle}} C_\lambda \right) \tag{4.34}$$

where $C_\lambda, \lambda_0 > 0$ are parameters and $\langle [q_{taker}(t) - \frac{1}{2}]^2 \rangle$ is determined by iterating $q_{taker}(t)$ for 10^6 time steps separately before the main simulation and obtaining the mean value of $[q_{taker}(t) - \frac{1}{2}]^2$. In essence, the above mechanism aims to mimic the risk mitigation behaviours adopted by liquidity providers in real financial markets. In cases in which the mid-price is not characterised by a strong trend and large price movements are therefore unlikely, liquidity providers tend to place limit orders close to the current mid-price to profit from small price fluctuations. In cases demonstrating a clear trend, however, which can occur in the model when $q_{taker}(t) \neq \frac{1}{2}$ and there is thus a tendency to buy as opposed to sell or vice versa, placing limit orders too close to the mid-price may result in rapid loses, leading liquidity providers to place orders at a greater depth (Preis et al. 2006; 2007). Additionally, the adoption of an exponential distribution as above results in a more realistic depth profile that more closely resembles those found in empirical order book data.

- Finally, all limit and market orders are of the same size, 1. Preis et al. (2006) ultimately find this change to have a negligible effect on the overall dynamics.

From the above, it should be apparent that the Preis et al. (2006) model builds upon the foundation of the work of Smith et al. (2003) in ways that result in increased qualitative consistency with empirical findings, while remaining, for the most part, a ZIA model. Indeed, most aspects of the model, including the submission rates of limit and market orders, as well as the decisions to buy or sell, remain entirely random. The model does, however, introduce some behavioural components, with the limit order placement mechanism having been adapted to reflect basic risk mitigation behaviours.

4.3.2.3 Chiarella and Iori (2002) Model

With two influential examples of the application of the ZIA paradigm now described, we conclude this section with the introduction of a model inspired by the work of Chiarella and Iori (2002), which follows a behavioural paradigm and attempts to directly simulate the decision making processes of trader agents.

As was the case for each of the previously considered models, the simulation progresses through a series of time steps, $t = 1, 2, \ldots, T$, with the mid-price being recorded at the end of each iteration. In contrast to the rate-based event generation approach adopted by the ZIA models, however, the approach of Chiarella and Iori (2002) does not model the flow of limit and market orders and order cancelation requests as independent Poisson processes. Rather, during each time step, t, one of the N_A trader agents in the current simulation, i, is selected at random and activates with probability λ. If agent i does activate, it follows a sequence of steps to determine its final trading decision:

- The trader agent begins by making a prediction regarding the spot return that will be observed over its investment time horizon, τ^i, which we define according to

$$\hat{r}_{t,t+\tau^i} = \frac{1}{g_1^i + |g_2^i| + n^i} \left[g_1^i \frac{1}{\tau_f} \frac{p_f - p_t}{p_t} + g_2^i \bar{r}_{\tau^i} + n^i \epsilon_t \right], \tag{4.35}$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$, p_t is the mid-price at time t, p_f is the constant fundamental value, $\tau_f > 0$ is the time horizon over which the asset price is expected to revert to its fundamental value, and

$$\bar{r}_{\tau^i} = \frac{1}{\tau^i} \sum_{j=1}^{\tau^i} \log \frac{p_{t-j}}{p_{t-j-1}} \tag{4.36}$$

is the observed one-period return averaged over the last τ^i time steps. As alluded to in Eqn. 4.35, trader i is considered to be following its own unique trading strategy, which is made up of, in order, fundamentalist, chartist, and noise trading components[36]. The unnormalised strategy weights associated with each trader are randomly drawn before the commencement of continuous trading and have distributions $g_1^i \sim |\mathcal{N}(0, \sigma_1^2)|$, $g_2^i \sim \mathcal{N}(0, \sigma_2^2)$, and $n^i \sim |\mathcal{N}(0, \sigma_n^2)|$. Since fundamentalists tend to be long-term investors and chartists day traders, the unnormalised strategy weights also determine the i-th agent's investment horizon,

$$\tau^i = \left\lceil \tau \frac{1 + g_1^i}{1 + |g_2^i|} \right\rceil, \tag{4.37}$$

where $\tau > 0$ is a model parameter defining a typical agent time horizon.

- Following the determination of the above spot return prediction, the i-th agent calculates a corresponding mid-price prediction according to

$$\hat{p}_{t+\tau^i}^i = p_t e^{\hat{r}_{t,t+\tau^i} \tau^i}. \tag{4.38}$$

If a price increase is predicted, the i-th agent will attempt to buy a single share of the asset, with the agent's desired bid price[37] being given by

$$p_b^i = \hat{p}_{t+\tau^i}^i (1 - k^i) \tag{4.39}$$

and the corresponding order being a limit order[38] with price p_b^i if $p_b^i < p_a$ and a market order otherwise. If a price decrease is expected instead, the i-th agent will attempt to sell a single share of the asset, with the agent's desired ask price being given by

$$p_a^i = \frac{\hat{p}_{t+\tau^i}^i}{1 - k^i} \tag{4.40}$$

and the corresponding order being a limit order with price p_a^i if $p_a^i > p_b$ and a market order otherwise[39]. Much like in the case of the unnormalised strategy weights, a k^i value is randomly drawn according to $U(0, k_{max})$, where $0 \leq k_{max} \leq 1$, for each trader agent before continuous trading commences.

[36] The Chiarella and Iori (2002) model can therefore be seen as sharing the same lineage as the interday models considered in Section 4.3.1.

[37] All order prices are, as in the Smith et al. (2003) model, rounded to the nearest multiple of the tick size, Δ.

[38] Limit orders remain in the order book until they are executed or until they are cancelled τ^i time steps after initially being placed.

[39] In this, the fundamental differences between the ZIA and behavioural paradigms become apparent. In the Chiarella and Iori (2002) model, the order type, decision to buy or sell, and the order price are derived from an application of a given agent's strategy to current market conditions and reflect the agent's attempt to profit from predicted price movements. This is in contrast to the mostly random behaviour of agents in models built on a ZIA foundation.

Following a total of T iterations of the above activation and trading routine, a corresponding series of T mid-prices would be obtained. Note, however, that during many of these iterations, no trading activity would have occurred and no mid-price changes would have been observed, since only a single agent is selected for activation during each time step and this agent may remain dormant with a probability of $1 - \lambda > 0$. Chiarella and Iori (2002) therefore construct a second, downsampled mid-price time series that acts as the final model output by sampling the prevailing mid-price every $N_{periods}$ time steps.

As a final remark, model initialisation is straightforward and involves the execution of $\lfloor 10\lambda N_{periods} \rfloor$ order placement events before the start of continuous trading, much like in the case of the previously considered ZIA models.

4.4 Results and Discussion

Having described our model selection procedure and each candidate model in detail, we now proceed with the presentation of our results, which we divide into two primary categories: exercises involving interday models and exercises involving intraday models.

4.4.1 Interday Models

As alluded to in Section 4.3.1, we consider a total of 6 approaches for modelling the daily log-return process, resulting in the model comparison set

$$\boldsymbol{M}_{inter} = \{\text{FW-HPM}, \text{FW-WP}, \text{FW-WH}, \text{ALW}, \text{ARMA-ARCH}, \text{ARMA-GARCH}\}, \quad (4.41)$$

where we have abbreviated the Franke and Westerhoff (2012) and Alfarano et al. (2008) models as FW and ALW respectively.

The model selection results associated with the above candidate set are further divided into two subsets, one related to pseudo-empirical validation exercises, as set out in Section 4.2.2, and another related to empirical applications involving three log-return time series associated with the closing prices of three major stock market indices. We now begin with a discussion of the former.

4.4.1.1 Simulation-based Validation Exercises

In our validation exercises, we consider two pseudo-empirical datasets, one generated according to the FW-HPM specification and another by the ARMA-ARCH model, with each data-generating process having been selected to ensure that the two pseudo-empirical datasets represent modelling approaches that are as distinct as possible[40]. The results of the above test cases, which are presented in Table 4.1, are ultimately very reassuring and indicate that both the DIC and PSIS-LOO are effective tools when used to discriminate between the considered interday models[41].

To elaborate, we notice that, firstly and most importantly, the true data-generating processes (the models that generated the pseudo-true data), are assigned the lowest DIC

[40] Our chosen values for $\boldsymbol{\theta}_{true}$ are presented in Table 4.5 in Appendix 4.A and we assume uniform priors over the explored parameter ranges (which are also specified in Table 4.5) when estimating a given model. Additionally, information related to the PSIS-LOO reliability diagnostics is available in Appendix 4.B.

[41] It is also worth mentioning that the DIC and PSIS-LOO estimates of out-of-sample predictive accuracy are quite similar, highly correlated (Pearson and Spearman correlation coefficients of 0.9997 and 0.9790 respectively), and imply the same ranking of the candidate models in almost all cases.

and PSIS-LOO estimates for both datasets, providing evidence that either model selection approach is effective when used to identify the true model. Additionally, we also notice that the relative ranking of each of the remaining models is largely intuitive, with models that are more conceptually similar to the true data-generating process being associated with lower criterion values. In the case of pseudo-empirical data generated by the FW-HPM model, we see that other variants of the Franke and Westerhoff (2012) model are assigned lower PSIS-LOO and DIC scores than the ALW model, which is itself assigned lower scores than the ARMA-ARCH and ARMA-GARCH models. Likewise, in the case of pseudo-empirical data generated by the ARMA-ARCH model, we observe that the ARMA-GARCH model is assigned a lower score than the remaining ABMs.

Table 4.1: Model Selection Result Summary for the Set of Interday Models Estimated on Pseudo-true Data

Model	DIC	PSIS-LOO	R (Likelihood)
FW-HPM	**−11933.9356**	**−11935.0728**	**1.0040**
FW-WP	−11921.7178	−11925.7841	1.0243
FW-WH	−11923.4073	−11927.8473	1.0172
ALW	−11891.6764	−11891.3286	1.0074
ARMA-ARCH	−11771.6239	−11768.8170	1.0096
ARMA-GARCH	−11797.3018	−11797.5859	1.0021
FW-HPM	−11764.6162	−11777.0316	1.0245
FW-WP	−11772.2927	−11775.9310	1.0118
FW-WH	−11790.3584	−11796.7848	1.0025
ALW	−11681.9759	−11688.8854	1.0186
ARMA-ARCH	**−12224.1700**	**−12223.8197**	**1.0002**
ARMA-GARCH	−12195.6238	−12194.4346	1.0012

The model that generated the pseudo-true data is indicated in bold. All PSIS-LOO values are multiplied by −2. Values of R (Gelman and Rubin 1992) less than 1.1 are typically viewed as indicative of convergence.

4.4.1.2 Empirical Applications

With the discriminative power of the proposed model selection tools now demonstrated, we proceed with a discussion of the results of our empirical applications. In these exercises, we consider the log-return time series derived from the closing prices of three major stock market indices, the FTSE 100 from 03-01-2012 to 30-12-2019, the Nikkei 225 from 01-03-2011 to 30-12-2019, and the DAX from 02-01-2012 to 30-12-2019, with the date ranges selected to ensure that the resulting empirical time series are both recent and of a similar length to the pseudo-empirical series used in the validation experiments[42].

The findings yielded by the application of our proposed protocol[43] to the above empirical datasets, which we present in Table 4.2, ultimately corroborate the conclusions of a number

[42]The FTSE 100 and Nikkei 225 datasets were previously considered in Section 3.3.2.

[43]The protocol is applied exactly as in the pseudo-empirical validation exercises, with the exception of the number of Markov chains generated for each candidate model, which we have increased from 5 to 10. Additionally, we no longer include the ARCH volatility specification, as GARCH is generally considered to be a more appropriate model for the volatility of actual return time series and the ARCH model was

of earlier investigations involving similar models. Firstly, Franke and Westerhoff (2012), who make use of moment-based techniques, find that the FW-HPM specification outperforms other variants of the Franke and Westerhoff (2012) model. Our approach leads to an analogous conclusion, with the HPM variant of the model being assigned lower PSIS-LOO and DIC scores than the WP and WH variants for all three datasets[44]. Secondly, both Barde (2016) and Bertschinger and Mozzhorin (2020), who employ information-theoretic and probabilistic programming methods respectively, conclude that the FW-HPM specification is capable of outperforming benchmark models that employ a GARCH volatility process and is, in many cases, the best performing ABM overall, an observation that is similarly supported by our results[45]. Finally, Bertschinger and Mozzhorin (2020), who consider the S&P 500 over two periods, January 2000 to December 2010 and January 2009 to December 2014, find that although the ALW specification performs competitively in the former period, its performance is relatively poor compared to a GARCH benchmark during the latter, suggesting that its overall competitiveness is inconsistent when compared to that of the FW-HPM specification[46]. Our results ultimately point to an identical conclusion, with the ALW model outperforming the ARMA-GARCH benchmark for the Nikkei 225 dataset, but underperforming in the remaining cases.

It therefore follows, in light of the above discussions, that our investigation has confirmed the competitiveness of the FW-HPM specification, relative to both an ARMA-GARCH benchmark and the remaining ABMs. A key limitation of the above analysis, however, is the fact that it deals only with the relative predictive power of the candidate models at the level of entire datasets. It would thus be interesting to delve deeper into the relative performances of the candidate models at a more granular level. Fortunately, as a byproduct of calculating the PSIS-LOO scores for entire datasets, we also obtain observation-level estimates of out-of-sample predictive accuracy, $\log p(\boldsymbol{x}_t|\boldsymbol{x}_{-t})$, which would facilitate such a discussion.

Now, with reference to Figure 4.1, which presents the aforementioned observation-level, out-of-sample log-likelihoods for each of the considered ABMs relative to the ARMA-GARCH benchmark[47], we observe a number of important trends. Firstly, we notice that, for both the FTSE 100 and Nikkei 225 datasets, the relative log-likelihood functions associated with the candidate ABMs tend to oscillate about the benchmark line for the majority of the considered time period, indicating that, for most observations, their performance is largely similar to that of the benchmark model. During certain periods, however, notably mid-2015 to mid-2016 for the FTSE 100 dataset and mid-2015 to late-2016 (and to a lesser extent early- to mid-2013) for the Nikkei 225 dataset, we observe a different trend in which the ABMs demonstrate a sustained and significant advantage relative to the benchmark. Interestingly, these periods are characterised by price movements that are far more extreme than in the remainder of the data, which seems to indicate that financial ABMs, as a class of models in general, have greater predictive performance when market conditions are sig-

originally included in the pseudo-empirical validation experiments to demonstrate the ability of the protocol to differentiate between competing ARMA-style models.

[44]It should be noted, however, that the DIC differences between the HPM, WH, and WP variants for the FTSE 100 dataset are less than 5, suggesting that distinguishing between these models is quite challenging in this case and that the predictive accuracy associated with each is likely very similar.

[45]We also find that the WP specification, though not considered by Barde (2016) or Bertschinger and Mozzhorin (2020), is competitive relative to the benchmark model in all three cases.

[46]The investigation of Barde (2016) similarly indicates the relative underperformance of the ALW specification relative to both the GARCH benchmark and FW-HPM specification.

[47]In all cases, we divide the considered time period into windows of length 50 and compute the average of the observation-level log-likelihoods for each window for improved clarity.

nificantly more volatile[48]. This is perhaps not surprising, however, given that the ABM tradition has long focused on the replication of return time series stylised facts, such as a fat-tailed distribution and volatility clustering, which would both be associated with the aforementioned volatile market events.

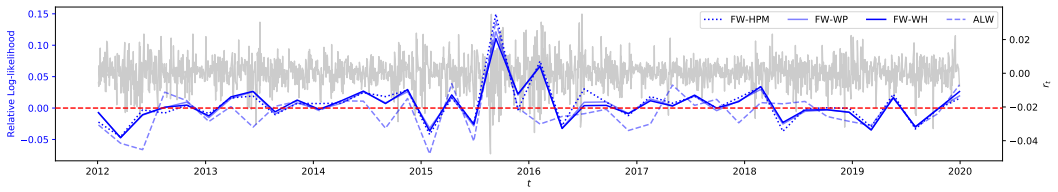
Table 4.2: Model Selection Result Summary for the Set of Interday Models Estimated on Empirical Data

Dataset	Model	DIC	PSIS-LOO	R (Likelihood)
FTSE 100	**FW–HPM**	**−13854.6474**	**−13856.6444**	**1.0821**
	FW–WP	−13851.9037	−13851.9202	1.0056
	FW–WH	−13852.5240	−13853.2040	1.0337
	ALW	−13810.8320	−13814.9080	1.0029
	ARMA–GARCH	−13833.1962	−13833.5180	1.0026
Nikkei 225	**FW–HPM**	**−13089.2151**	**−13087.8728**	**1.0267**
	FW–WP	−13074.9539	−13074.4329	1.0130
	FW–WH	−13072.63316	−13071.6728	1.0020
	ALW	−13077.8963	−13075.2678	1.0058
	ARMA–GARCH	−13009.7141	−13011.1859	1.0051
DAX	**FW-HPM**	**−12761.1559**	**−12760.9447**	**1.0047**
	FW–WP	−12755.1923	−12753.7160	1.0372
	FW–WH	−12746.2281	−12746.2108	1.0198
	ALW	−12745.7336	−12749.3666	1.0149
	ARMA–GARCH	−12753.9589	−12752.9391	1.0043

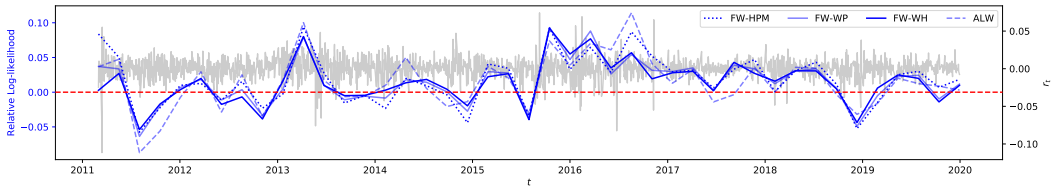
The best performing model is indicated in bold. All PSIS-LOO values are multiplied by −2. Values of R (Gelman and Rubin 1992) less than 1.1 are typically viewed as indicative of convergence.

Of course, this does not imply that this is necessarily true during any sustained increase in volatility, however. Indeed, Figure 4.1c, which presents the corresponding analysis for the DAX, requires a more nuanced discussion. While the DAX dataset does, much like the FTSE 100 and Nikkei 225 datasets, contain a sub-period characterised by heightened volatility (mid-2015 to early-2016), the price dynamics observed during this period are noticeably less extreme (relative to the price movements observed in the remainder of the data) when compared to the dynamics characterising similar periods in the previously considered datasets. This small, but important nuance appears to have relevant implications for the relative performances of the considered models. Although the observation-level log-likelihood functions do once again demonstrate oscillatory behaviour across the majority of the dataset, most of the ABMs no longer consistently outperform the benchmark during the aforementioned period of sustained heightened volatility. Rather, we find that, even though all of the considered ABMs exhibit increases in their relative log-likelihood functions during this period, most also demonstrate at least one sharp decrease in out-of-sample predictive accuracy that renders them largely uncompetitive. Interestingly, it appears that only the FW-HPM model exhibits a consistent advantage over the benchmark in this case, having

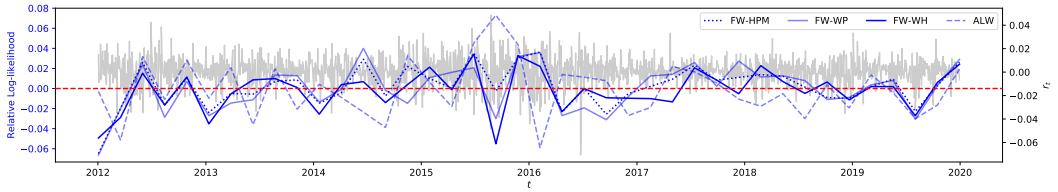
[48]This is also in line with the findings of Barde (2016), who employs a different observation-level metric.



(a) FTSE 100



(b) Nikkei 225



(c) DAX

Figure 4.1: Observation-level log-likelihoods (left axis) for the set of interday ABMs, shown relative to the benchmark ARMA-GARCH model (dotted red line) and overlaying the observed log-returns (right axis) to which the models were calibrated.

a positive relative log-likelihood for the majority of this period. This ultimately seems to suggest that although ABMs may deliver superior performance during sustained periods of significantly increased volatility, most of the considered specifications, with the exception of the HPM variant of the Franke and Westerhoff (2012) model, appear less robust in the case of sustained, but less dramatic increases[49].

4.4.2 Intraday Models

In Section 4.4.1, we provided a detailed study of the application of our proposed model selection protocol to a number of simple interday financial ABMs. In these exercises, we demonstrated the protocol's ability to effectively discriminate between a number of candidate models, both explicitly through a series of validation exercises involving pseudo-empirical data, and implicitly through an empirical study that yielded findings consistent with those

[49]It should, nonetheless, be noted that ABMs are frequently motivated as a response to issues regarding the applicability of traditional models to unstable market periods, such as the Great Recession of the late 2000s (Geanakoplos and Farmer 2008; Farmer and Foley 2009; Fagiolo and Roventini 2017). It thus follows that this improvement in performance within extreme market regimes acts as compelling evidence for the adoption of the ABM paradigm to model such market conditions, even if some ABM specifications may not exhibit similar improvements in predictive performance during more benign periods.

obtained in previous investigations using different tools. This ultimately provides strong support for the general applicability of our proposed approach.

It should nonetheless be clear, however, that the previously considered models are not the true focus of our investigation and instead act primarily as a test case to establish support for the overall methodology, despite yielding a number of interesting insights in their own right. We thus progress to a second and more challenging set of candidate models,

$$\boldsymbol{M}_{intra} = \{\text{PGPS}, \text{SFGK}, \text{CI}, \text{ARMA-GARCH}\}, \tag{4.42}$$

where we have abbreviated the Preis et al. (2006) model as PGPS, the Smith et al. (2003) model as SFGK, and the Chiarella and Iori (2002) model as CI.

As alluded to throughout this chapter, the three primary models being studied above (PGPS, SFGK, and CI) possess a number of traits that limit the applicability of other model selection approaches that may have been previously applied to models of the type contained in the set $\boldsymbol{M}_{inter}$. These include, but are not limited to, significant computational costs, and high-dimensional, discrete latent spaces represented as central LOBs. Fortunately, our approach remains applicable, even when considering models of this class, allowing us to go beyond what has been attempted in most of the existing literature.

4.4.2.1 Simulation-based Validation Exercises

Table 4.3: Model Selection Result Summary for the Set of Intraday Models Estimated on Pseudo-true Data

Model	DIC	PSIS-LOO	R (Likelihood)
PGPS	**−24078.4837**	**−24074.2981**	**1.0138**
SFGK	−23975.8589	−23975.8606	1.0007
CI	−23858.1606	−23873.4189	1.0357
ARMA–GARCH	−23659.0571	−23650.5487	1.0015
PGPS	−24083.9940	−24084.5450	1.0851
SFGK	−24119.7391	−24120.4082	1.0094
CI	**−24233.2726**	**−24232.3925**	**1.0093**
ARMA–GARCH	−23933.2822	−23930.7311	1.0044

The model that generated the pseudo-true data is indicated in bold. All PSIS-LOO values are multiplied by −2. Values of R (Gelman and Rubin 1992) less than 1.1 are typically viewed as indicative of convergence.

As before, we begin with a simulation-based validation exercise and consider two pseudo-empirical log-return series, one generated by the PGPS specification and another by the CI model[50].

The results of these exercises, which we present in Table 4.3, ultimately indicate that the performance of the proposed model selection protocol is not affected by the increased sophistication of the candidate models, with the true data-generating processes being assigned the

[50] Our chosen values for $\boldsymbol{\theta}_{true}$ and the explored parameter ranges are presented in Table 4.6 in Appendix 4.A.

lowest DIC and PSIS-LOO estimates in all cases and the overall model rankings remaining intuitive[51]. To elaborate, we find that, for the set of exercises involving pseudo-empirical data generated by the PGPS ZIA model, the SFGK model, which is also an example of the ZIA paradigm, is ranked second, the CI model, which adopts a behavioural paradigm but retains a LOB structure, is ranked third, and the ARMA-GARCH model, which is neither agent-based nor explicitly incorporates any order book structure, is ranked last. Similarly, for the set of exercises involving pseudo-empirical data generated by the CI model, we find that the remaining LOB models are predicted to be more appropriate than the ARMA-GARCH model, as we would expect.

4.4.2.2 Empirical Applications

With the effectiveness of our proposed model selection protocol now demonstrated in the more challenging intraday context, we proceed with an empirical application in which we calibrate each model in our set of candidates to order book data for three liquid stocks listed on the London Stock Exchange (LSE), including Vodafone PLC, Barclays PLC, and Tesco PLC[52]. As a result of the increased sophistication of the datasets being considered, a number of preprocessing steps are required before our protocol may be applied. We outline these and other details regarding our considered empirical data in Appendix 4.C.

Table 4.4: Model Selection Result Summary for the Set of Intraday Models Estimated on Empirical Data

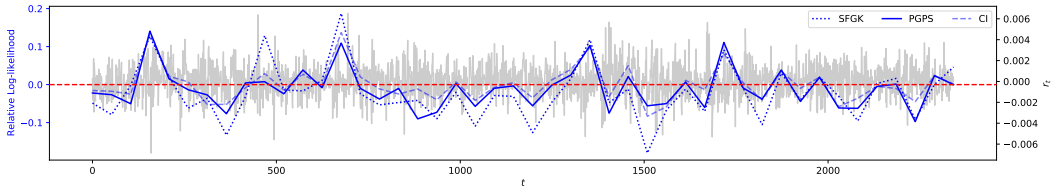
Dataset	Model	DIC	PSIS-LOO	R (Likelihood)
VODA	PGPS	−23964.1244	−23964.8101	1.0257
	SFGK	−23905.0644	−23912.6745	1.0009
	CI	**−24010.3003**	**−24014.1787**	**1.0319**
	ARMA–GARCH	−24009.6426	−24010.0465	1.0021
BARC	PGPS	−23602.6653	−23610.9456	1.0226
	SFGK	−23512.1757	−23518.7243	1.0123
	CI	**−23683.7940**	**−23684.6918**	**1.0513**
	ARMA–GARCH	−23656.9053	−23656.4985	1.0056
TESC	PGPS	−23762.0342	−23759.7089	1.0489
	SFGK	−23693.8324	−23701.2011	1.0252
	CI	**−23810.0719**	**−23821.0107**	**1.0184**
	ARMA–GARCH	−23758.7489	−23766.2268	1.0027

The best performing model is indicated in bold. All PSIS-LOO values are multiplied by −2. Values of R (Gelman and Rubin 1992) less than 1.1 are typically viewed as indicative of convergence.

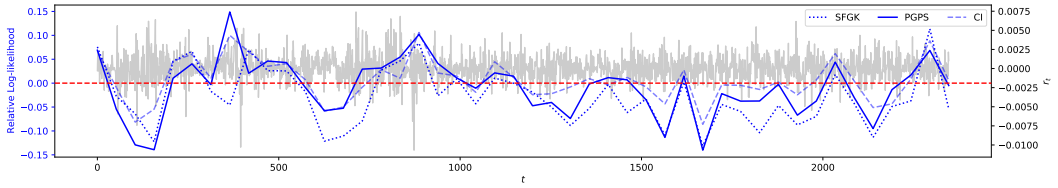
Proceeding to a discussion of the obtained results, we refer to Table 4.4, which presents the main findings of the above empirical analysis. First and foremost, we observe that the CI model is, in all cases, assigned the lowest DIC and PSIS-LOO scores. For the Barclays and Tesco datasets, we find these scores to be substantially lower than those associated

[51]It is also worth noting that the DIC and PSIS-LOO estimates of out-of-sample predictive accuracy are again quite similar and imply an identical ranking of the candidate models throughout.

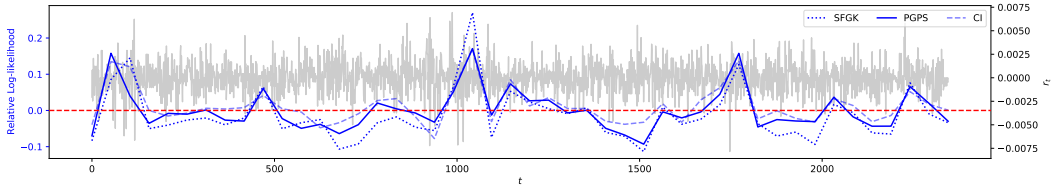
[52]For all stocks, we consider the period beginning on 01-10-2019 and ending on 30-11-2019.



(a) Vodafone PLC



(b) Barclays PLC



(c) Tesco PLC

Figure 4.2: Observation-level log-likelihoods (left axis) for the set of intraday ABMs, shown relative to the benchmark ARMA-GARCH model (dotted red line) and overlaying the observed log-returns (right axis) to which the models were calibrated.

with the benchmark ARMA-GARCH model, suggesting significant outperformance. We should point out, however, that, when considering the Vodafone dataset, the DIC and PSIS-LOO differences are too small to make a definitive conclusion. Nevertheless, when viewed collectively, the obtained results do indeed provide compelling evidence that the CI specification is competitive. Unfortunately, the same cannot be said of the PGPS and SFGK specifications. Overall, we find that, in most cases, both of the considered ZIA models tend to underperform, producing noticeably larger DIC and PSIS-LOO scores when compared to the benchmark.

Figure 4.2, which presents the out-of-sample, observation-level log-likelihoods for each dataset[53], provides a more detailed examination of the phenomena described above. To elaborate, we observe a somewhat analogous trend to what we encountered in the interday case in which ABMs appear to perform best during more volatile periods or near extreme events (particularly negative ones), while tending to either behave similarly or underperform relative to the benchmark during quieter periods[54]. Probing more deeply, we observe that although all of the considered ABMs tend to peak in performance and deliver predictions of

[53] We again consider averages over windows of length 50 for improved clarity.

[54] As in the interday case, we also find this trend to be more pronounced and clearly identified in two of the datasets, specifically Barclays and Tesco, and less obvious for the Vodafone dataset.

a similar quality during the same extreme periods[55], the behaviour of the CI specification is quite different to that of the ZIA models during more benign periods. In particular, we observe that the ZIA models tend to experience sharper drops in performance, often significantly underperforming relative to the benchmark, while the CI specification tends to produce predictive likelihoods that remain closer to the benchmark line. This seems to be a major factor underpinning the CI specification's success in this context, with the model being flexible enough to perform well during volatile periods that prove difficult for standard time series models, while still remaining reasonable during other, less extreme periods.

It thus follows that our overall conclusions are similar to those presented in the interday case. In short, ABMs (as a general class of models) appear to perform well during extreme periods, regardless of the exact approach taken to construct them, which, as previously stated, is not surprising given their emphasis on recovering stylised facts such as fat-tailed return distributions and volatility clustering. It would seem that what truly defines an effective ABM, however, is the ability to remain competitive during less volatile periods.

Ultimately, it is difficult to establish whether the observed performance differences can be attributed to the fact that the CI model adopts a behavioural as opposed to ZIA approach, since definitive conclusions would require the consideration of larger sets of behavioural and ZIA models, which is significantly beyond the scope of this thesis. It is worth pointing out, however, that we have still nonetheless provided an initial attempt at directly comparing the paradigms and demonstrated that the CI specification appears to produce more realistic intraday return dynamics than the SFGK and PGPS models, which, to the best of our knowledge, was not known previously. We therefore feel that exercises of the type considered here could help play an important role in selecting which models and paradigms would be most worth building upon in future work.

Conclusion

In the preceding sections, we have demonstrated how the Bayesian estimation methodology introduced in Chapter 3 may be combined with the DIC and PSIS-LOO to produce a widely-applicable protocol for Bayesian model selection in the context of ABMs that allows for the consideration of both simple, interday financial models and more sophisticated, computationally-expensive intraday variants. In all cases, we find that, when underpinned by our proposed neural network-based likelihood approximation, both the DIC and PSIS-LOO produce sensible results in exercises involving pseudo-empirical data. Additionally, both criteria are highly-correlated and lead to identical model rankings in all but a minority of cases. This suggests that both approaches to model selection are ultimately reasonable choices in this context, despite individual pros and cons.

When applying our proposed protocol to return time series data drawn from the closing prices of three major stock market indices, we observe a great degree of consistency with the existing literature. As an example, we demonstrate the superiority of the DCA-HPM version of the Franke and Westerhoff (2012) model relative to the DCA-WP and DCA-WH variants and additionally showcase its ability to outperform an ARMA-GARCH benchmark. Further, we confirm that the competitiveness of the DCA-HPM specification stems not from consistent outperformance relative to the benchmark for the majority of the considered time period, but rather from a tendency to perform equivalently-well during less extreme

[55] See, for example, bin 200 of the Vodafone dataset, bins 400 and 800 of the Barclays dataset, and bins 1000 and 1750 of the Tesco dataset.

market periods and produce significantly more accurate predictions during times of stress, i.e. heightened volatility.

Moving beyond the existing literature, we also apply our protocol to intraday return time series constructed using downsampled order book data for three liquid, LSE-listed stocks in order to provide a first attempt at comparing members of the competing ZIA and behavioural LOB modelling approaches. Overall, we find that the behavioural model introduced by Chiarella and Iori (2002) delivers superior performance relative to an ARMA-GARCH benchmark and, as in the case of the DCA-HPM version of the Franke and Westerhoff (2012) model, seems to do so as a result of more or less matching the benchmark's performance during quieter periods and greatly exceeding it in periods characterised by more extreme price movements. In contrast to this, we find that both of the considered ZIA models, specifically that of Preis et al. (2006) and a variant of the Smith et al. (2003) model introduced by Preis et al. (2007), frequently underperform relative to the benchmark. Much like the Chiarella and Iori (2002) model, we find that the considered ZIA models do indeed deliver relatively accurate predictions in periods characterised by very large price movements. For the remainder of the considered period, however, we find that both experience sharp decreases in predictive accuracy, which seems to explain why both ZIA models tended to be ranked poorly at the aggregate level by the DIC and PSIS-LOO.

As a final remark, it must be noted that we are by no means suggesting that the above conclusions regarding the considered intraday models are definitive. Rather, our investigation instead acts as a first attempt to quantitatively compare such models, which, to the best of our knowledge, is an area of inquiry that has largely been ignored in the past. It would certainly be worthwhile to, in future work, consider a larger pool of ZIA and behavioural models and explore data from a wider variety of exchanges and time periods to reach a more definitive conclusion. Nevertheless, we do believe that we have demonstrated two important points: the utility of our proposed protocol for comparing ABMs, even when relatively complex and computationally-expensive, and the value of model selection exercises of this nature in general in helping discriminate between models that may seem equally compelling at face value[56].

[56]Though not a major focus of this thesis, we also showcase, in Appendix 4.D, how our neural network-based approach may be used to facilitate the process of making actual out-of-sample forecasts. Additionally, we also provide further evidence to support the main conclusions of this chapter.

4.A Parameter Priors and Pseudo-empirical Data Parameterisation

Table 4.5: Explored Parameter Ranges and Pseudo-true Parameter Values for the Set of Interday Models

	Parameter	θ_l	θ_u	θ_{fixed}	θ_{true}
FW					
	μ	–	–	0.01	0.01
	β	–	–	1	1
	ϕ	0	5	–	0.15
	χ	0	5	–	1.5
	α_0	-2	2	–	-1.5
	α_n	0	4	–	3.5
	α_p	0	40	–	18.5
	σ_f	0	5	–	0.75
	σ_c	0	5	–	2
	α_w	0	10000	–	–
	η	0	1	–	–
	p^*	–	–	0	0
ALW					
	N	–	–	100	–
†	a	0	5	–	–
†	b	0	5	–	–
†	σ_f	0	300	–	–
ARMA					
	a_1	-1	1	–	0.15
	a_2	-1	1	–	0.15
	b_1	-1	1	–	0.2
	b_2	-1	1	–	0.2
†	c_0/ω	0	1	–	0.5
†	c_1/α_1	0	1	–	0.5
†	c_2	0	1	–	0.5
	β_1	0	1	–	–

† indicates a parameter for which the associated values have been multiplied by 10^4.

In Footnote 40, we briefly indicated that we employ uniform priors in all of our Bayesian model selection experiments. To be more precise, we consider a prior of the form

$$p(\theta) \sim U(\theta_l, \theta_u), \tag{4.43}$$

for each free parameter[57], θ, in each Bayesian estimation run. In this case, $[\theta_l, \theta_u]$ represents the explored parameter range, which we specify for each interday and intraday model parameter in Tables 4.5 and 4.6 respectively[58]. Additionally, if a given model is also used to generate pseudo-empirical data for use in our validation exercises, the associated parameter values are also specified in the above tables in the column labelled θ_{true}.

Table 4.6: Explored Parameter Ranges and Pseudo-true Parameter Values for the Set of Intraday Models

	Parameter	θ_l	θ_u	θ_{fixed}	θ_{true}
PGPS					
	N_A	–	–	125	125
	α	0.15	0.75	–	0.25
	μ	0	0.15	–	0.1
	δ	0	0.1	–	0.04
	Δ_S	0	0.1	–	0.035
	λ_0	0	300	–	100
	C_λ	0	30	–	10
	p_{scale}	–	–	100	100
SFGK					
	α	10	50	–	–
	μ	0	10	–	–
	δ	0	0.1	–	–
	p_{lim}	0	25000	–	–
	Δ	–	–	0.01	0.01
CI					
	N_A	–	–	1250	1250
*	τ	1	3	–	2.2
*	τ_f	–	–	200	200
	λ	0	1	–	0.5
†	σ_ϵ	0.01	1	–	0.1
◊	σ_1	0	10	–	2
◊	σ_2	0	10	–	10
◊	σ_n	0	1	–	0.2
	k_{max}	0	0.05	–	0.015
	p_f	–	–	1000	1000
	Δ	–	–	0.01	0.01
	$N_{periods}$	–	–	50	50

† indicates a parameter whose values have been multiplied by 10^4. * indicates measurement in units of $N_{periods}$. ◊ indicates measurement in units of σ_ϵ. $p_0 = 1000$ for all models.

[57] A small number of model parameters are not estimated but instead fixed to a-priori selected values, θ_{fixed}. This is done for one of three reasons: a given parameter is redundant and simply rescales other parameters (N, N_A, p_{scale}, τ_f, μ, β), is fixed and known for a given market (Δ), or is set to be fixed by the original model authors (p_f, p^*).

[58] These parameter ranges are employed for the simulation-based validation exercises as well as the empirical applications.

4.B PSIS-LOO Reliability Diagnostics

When introducing PSIS-LOO in Section 4.1.3, we indicated that the final criterion value is a sum of contributions associated with each empirical observation. For each of the aforementioned observations, we obtain a diagnostic, $\hat{k}_t, t = 1, 2, \ldots, T_{emp}$, which can be used to determine the reliability of the PSIS-LOO contribution associated with that observation. As previously stated, values of $\hat{k}_t < 0.5$ indicate rapid convergence via the central limit theorem, values of $\hat{k}_t > 1$ indicate that no version of the central limit theorem applies and that the estimate is therefore unreliable, and $0.5 \leq \hat{k}_t \leq 1$ indicates that a generalised version of the central limit theorem applies, though convergence may be slow, especially as $\hat{k}_t$ approaches 1. Given these theoretical findings, it is very clear that $\hat{k}_t$ should never exceed 1 and that values of $\hat{k}_t < 0.5$ almost certainly suggest reliability. The case where $0.5 \leq \hat{k}_t \leq 1$ is, however, not immediately clear. In response, Vehtari et al. (2017) conduct a series of experiments and suggest a cutoff of approximately 0.7 for diagnosing the reliability of a given observation-level contribution, which we adopt as a baseline in our experiments.

Overall, the PSIS-LOO reliability diagnostics obtained in our entire set of model selection experiments, both inter- and intraday, can be summarised as follows:

- No $\hat{k}_t$ value ever exceeds 1.
- For the vast majority of model and dataset combinations (over 85%), every $\hat{k}_t$ value is additionally below 0.7.
- For the remaining model and dataset combinations, only a very small number of observations (0.5% of the total number of observations in the worst case) are associated with values of $\hat{k}_t$ larger than 0.7. In all of these cases, we also observe that $\hat{k}_{max} = \max_{t \in \{1,2,\ldots,T_{emp}\}} \hat{k}_t$ is still relatively close to 0.7. To illustrate, the 3 largest $\hat{k}_{max}$ values obtained in our experiments are 0.8123, 0.7683, and 0.7407.
- For each example (model and dataset combination) originally yielding $\hat{k}_t$ values exceeding the reliability threshold of 0.7, we also conducted a check in which we recomputed the PSIS-LOO scores and diagnostics using a longer thinning interval[59] of 10 sample sets as opposed to 5. Overall, we found that, for all but one model and dataset combination, this change resulted in virtually identical PSIS-LOO scores and all $\hat{k}_t$ values now being below 0.7, providing support for the reliability of the PSIS-LOO scores in these cases.

Based on the above summary, we argue that it is reasonable to conclude that our obtained PSIS-LOO scores are, on the whole, likely to be reliable, especially given that 0.7 is an approximate threshold. Additionally, it should be noted that we also include the DIC as an alternative model selection tool in our experiments and find that, in almost all cases, identical conclusions are yielded to those associated with PSIS-LOO.

4.C Order Book Data Preprocessing

The fitting of financial ABMs to empirical data at intraday time-scales introduces a number of additional difficulties not encountered in the interday case. First and foremost, while daily

[59] See Footnote 16 for more details. In essence, we are considering every second set of observation-level likelihoods in the original thinned chain.

data is widely and freely available, most intraday data is proprietary and requires a licence from a vendor, which may, in some cases, be relatively expensive. Secondly, the process of ABM estimation, at least as we have defined it throughout this thesis, requires data to be presented as a time series. In contrast to this, most empirical intraday data is presented in an event-based or tick-by-tick format[60] and each data point represents an entire LOB state consisting of a large number of limit orders, necessitating some form of downsampling. We therefore proceed with a discussion on how we obtained the required LOB data and further introduce our approach to transform it into an appropriate format.

4.C.1 BMLL Data Lab

All of the required LOB data was provided to us courtesy of BMLL Technologies and accessed via the Data Lab (BMLL 2021) platform, which provides an API to interact with and preprocess order book data. The data in question is presented as a series of LOB states for a given trading day, or in other words, a series of objects containing all of the orders present in the order book at the time the state was recorded. Crucially, this series is event-based in nature: a new data point is added to the series only after an event occurs that would change the state of the order book, such as a trade (i.e. orders being matched and executed), the addition of a new limit order, or the cancellation of an existing one. In addition to the sequence of LOB states itself, the platform also provides methods to determine what kinds of events led to the aforementioned state changes, and, importantly, the volume if the event was a trade.

Although access to a wide variety of major exchanges is supported by the service, we were given access to LSE data exclusively and thus consider three liquid, LSE-listed stocks, including Vodafone Group PLC, Barclays PLC, and Tesco PLC, over the two-month period[61] beginning on 01-10-2019 and ending on 30-11-2019.

For each trading day considered, it is important to note that the obtained datasets do not only represent the continuous trading period, 8:00 to 16:30, but also include data related to an opening auction occurring from 07:50 to 8:00, a midday auction occurring from 12:00 to 12:02, and a closing auction occurring from 16:30 to 16:35. During each of these auctions, alternative trading regimes are active and these periods should therefore be excluded when performing our analysis. It follows then that, when considering the data for a given trading day, we remove any order book state data points sampled outside of the periods occurring[62] from 8:30 to 12:00 and 12:07 to 16:30 before applying our downsampling strategy, in line with standard procedures (Platt and Gebbie 2018; Ntakaris et al. 2018).

4.C.2 Downsampling

When downsampling the above data, we note that, in our analysis, we are primarily concerned with determining which models produce simulated data that most accurately mimics the intraday dynamics of mid-price returns, or equivalently, which models best explain intraday price movements and the associated volatility thereof. This should be contrasted with an analysis of how well a given model replicates the overall features of the microstructure

[60] In other words, the time between each measurement is not constant.

[61] This strikes a balance between being sufficiently long such that we are able to consider a reasonable sample of different trading days, while still being short enough to remain computationally-tractable.

[62] Note that, after both the opening and midday auctions, we wait for a brief period before resuming our consideration of the order book state in order to allow the market to stabilise and fully re-enter the continuous trading regime.

of the market, such as bid-ask spreads, depth profiles, and so on, which we do not consider here[63]. It thus follows that the primary aim of the proposed downsampling procedure is the construction of a mid-price time series.

The simplest strategy for achieving the above involves the construction of price bars that correspond to some fixed calendar time interval. As an example, one might divide the considered trading day into a series of bins corresponding to 1 minute of trading time each and set the price associated with each bin to be the final observed mid-price in that bin. Such an approach is considered in the work of Platt and Gebbie (2018), which provides an early attempt at the use of downsampled mid-price data to calibrate an intraday ABM.

Unfortunately, a key weakness of the above downsampling strategy lies in the fact that differing levels of trading activity may prevail in each bin: some bins may contain little or no trading activity and thus be uninformative, while others may, in contrast, correspond to particularly busy periods and thus not be sampled at an appropriately high frequency. Fortunately, a more appropriate alternative to the above, known as volume time (Easley et al. 2012), is available and ensures that each recorded datapoint is associated with a similar level of trading activity, leading it to become our approach of choice.

Under such a scheme, the data is divided into N_{bins} bins, each of which corresponds to a roughly identical trading volume,

$$V_{bin} = \frac{V_{total}}{N_{bins}}, \tag{4.44}$$

where V_{total} is the total trading volume for the entire dataset. More concretely, we assign each consecutive LOB update to the current bin until the volume of trades in the bin is equal to or exceeds V_{bin}, at which point we move on to the next bin and continue until we have moved through the entire dataset. As in the case of the calendar time scheme, the price associated with each bin is set to be the final observed mid-price in that bin, yielding our final dataset.

4.C.2.1 Considering Multiple Trading Days

Applying the above scheme to data from a single day of continuous trading is a relatively straightforward exercise. When investigating longer periods as we do in this investigation, however, we end up needing to consider data from multiple trading days and this introduces a number of important nuances. To elaborate, a naive approach might involve the application of a volume time downsampling strategy to each trading day and the concatenation of the resultant mid-price bars to form a single time series. This can be problematic, as it assumes a single price movement from the end of one trading day to the beginning of the next. Since each trading day begins and ends with opening and closing auctions respectively, during which alternative trading regimes are active and a large amount of trading activity may be present, a naive concatenation strategy may lead to atypical price movements between trading days, which, in turn, could lead to outliers being present in the final return time series to which we calibrate the model.

In response, we propose a more theoretically-sound alternative. In more detail, we assume that the return time series derived from the downsampled mid-prices for a given trading day, $i = 1, 2, \ldots, I$, is denoted by $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,T_i}]$, and, much like in

[63]It should nevertheless be stated that such features likely do have a profound effect on price movements and models which more accurately replicate other features of a given market's microstructure are more likely to produce realistic mid-price movements.

the machine learning literature, represents a single data case in a larger batch of training examples[64], $\boldsymbol{X}$. This implies an updated definition of the likelihood first introduced in Eqn. 3.9:

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{i=1}^{I} p(\boldsymbol{X}_i|\boldsymbol{\theta}) \tag{4.45}$$

$$= \prod_{i=1}^{I} \prod_{t=1}^{T_i-L} \tilde{f}(\boldsymbol{x}_{i,t}, \ldots, \boldsymbol{x}_{i,t+L-1}, \boldsymbol{x}_{i,t+L}, \boldsymbol{\phi}). \tag{4.46}$$

It follows then that the resultant full-period likelihood can, in essence, be viewed as a composite likelihood built from the product of individual, single-day likelihoods.

Implicitly, this construction assumes that each trading day is independent of those before it. While some may initially feel that such an assumption may not be ideal, we nonetheless view it as reasonable. This is because the effects of the opening and closing auctions are significant and can more or less be viewed as initiating a new continuous trading regime each day. Indeed, this is why naive concatenation approaches are problematic in the first place and we thus believe that this ultimately warrants the treatment of each trading day as a separate entity.

4.C.3 Illustrations

In line with the preceding discussions, constructing our final collection of datasets is now a straightforward process. For each LSE-listed stock, we proceed by:

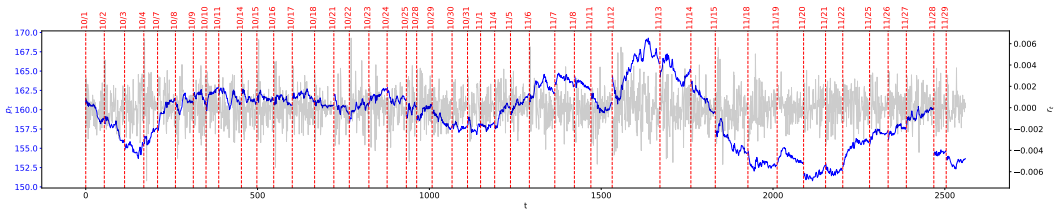
- Gathering the data for all trading days present in the period beginning on 01-10-2019 and ending on 30-11-2019. For each of these days, we remove all LOB states occurring outside of the periods 8:30 to 12:00 and 12:07 to 16:30.
- Calculating the total trade volume across all of the trading days, V_{total}, and using this to determine the bin size, V_{bin}, for $N_{Bins} = 2500$.
- Constructing a set of volume time mid-price bars for each trading day, with each bar having size V_{bin}. In general, each trading day will yield a different number of data points, but each data point will correspond to a similar volume of trade having occurred since the last measurement.
- Calculating a log-return time series for each trading day using the volume time mid-prices. Each return time series corresponds to a single instance of $\boldsymbol{X}_i$.

The results of the above procedure are presented in Figure 4.3. As is evident from the illustrations, large mid-price jumps can occur from the end of one trading day to the beginning of the next, making naive concatenation-based approaches inappropriate and further highlighting the necessity of our preprocessing approach.

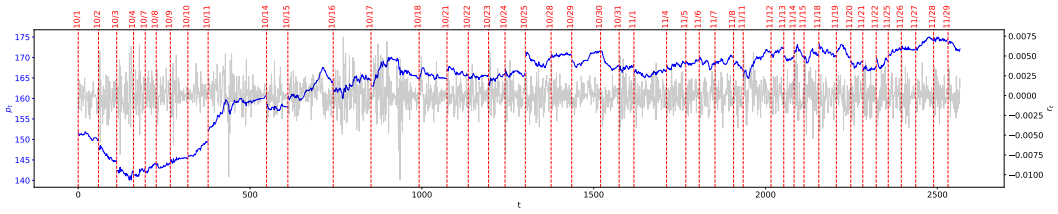
4.D Bayesian Forecasting

With clear strategies for parameter estimation and model selection now described, we proceed to briefly discuss how the techniques introduced throughout this thesis may also be

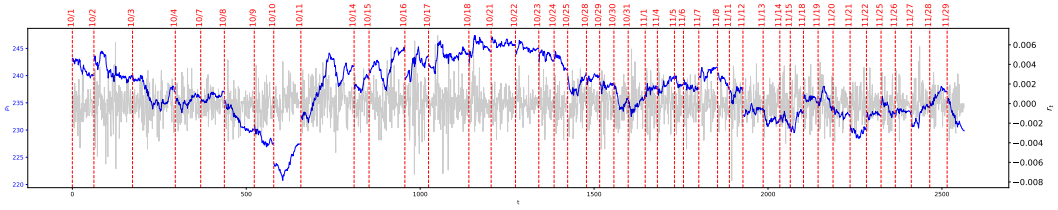
[64] It should be noted that the same volume bin size should be used for each day.



(a) Vodafone PLC



(b) Barclays PLC



(c) Tesco PLC

Figure 4.3: Downsampled mid-price, p_t, and log-return, r_t, time series for three LSE-listed stocks for the period beginning on 01-10-2019 and ending on 30-11-2019. Each dashed red line represents the start of a new trading day and each day is treated as a separate entity.

used to facilitate the computation of the posterior predictive distribution and thus yield a straightforward and widely-applicable approach to fully-Bayesian time series forecasting.

Though our approach differs in a number of key areas, it draws significant inspiration from the methodology proposed by Delli Gatti and Grazzini (2020). To elaborate, their methodology proceeds by:

- generating a set of model Monte Carlo replications, $\left[\boldsymbol{x}_{1,i}^{sim}(\hat{\boldsymbol{\theta}}), \boldsymbol{x}_{2,i}^{sim}(\hat{\boldsymbol{\theta}}), \ldots, \boldsymbol{x}_{T_{sim},i}^{sim}(\hat{\boldsymbol{\theta}})\right]$, for a single point estimate[65], $\hat{\boldsymbol{\theta}}$, that is obtained by calibrating the model of interest to a set of in-sample data, $\boldsymbol{X}$,
- using these Monte Carlo replications to learn the conditional density, $p(x_t^{sim}|x_{t-1}^{sim}, \hat{\boldsymbol{\theta}})$, of observations simulated by the model[66],
- and finally generating one-step ahead forecasts by calculating[67] $\mathbb{E}[x_{t+1}^{sim}|\tilde{x}_t, \hat{\boldsymbol{\theta}}]$ for each

[65] Delli Gatti and Grazzini (2020) consider a MAP estimate with uniform priors, which essentially corresponds to the maximum likelihood estimate.

[66] This is achieved through a simple KDE-based procedure.

[67] A simple discretisation scheme is applied.

(future) out-of-sample observation, $\tilde{x}_t \notin \boldsymbol{X}$.

In our case, however, and in addition to a number of other nuanced differences, we do not consider a single point estimate, but rather integrate over the entire parameter posterior, $p(\boldsymbol{\theta}|\boldsymbol{X})$. This allows any forecasts made to take the uncertainty associated with the estimation procedure into account and thus, as we will soon show, be significantly more accurate.

4.D.1 An Illustrative Example: The S&P 500

Having now introduced our inspiration, we proceed with a more detailed exposition of our own procedure, which we believe is best described through an illustrative example.

In this case, we begin by applying the protocol employed in Section 4.4.1.2 to the log-return time series derived from the closing prices of the S&P 500 from 01-01-1986 to 01-01-2006 to sample the associated parameter posterior of the model of interest[68]. Then, recall, from Section 2.1.3.4, that the final sample set obtained using the employed Metropolis-Hastings algorithm is distributed according to the aforementioned parameter posterior, $p(\boldsymbol{\theta}|\boldsymbol{X})$. We therefore generate a set of $R = 50$ model Monte Carlo replications of simulated (rebased) log-prices[69],

$$x_{t,i}^{sim}(\hat{\boldsymbol{\theta}}_n) = \log \frac{P_{t,i}^{sim}(\hat{\boldsymbol{\theta}}_n)}{P_{1,i}^{sim}(\hat{\boldsymbol{\theta}}_n)}, t = 1, 2, \ldots, T_{sim} = 2000, i = 1, 2, \ldots, R = 50, \tag{4.47}$$

for each of the $n = 1, 2, \ldots, N = 70$ parameter samples in this final sample set, where $P_{t,i}^{sim}$ is the simulated asset price at time t for random seed i. This is then followed by the application of the training procedure discussed in Chapter 3 to learn separate MDNs for each of the N sets of Monte Carlo replications, which ultimately yields a set of N conditional density estimators denoted by

$$\tilde{p}(x_t^{sim}|x_{t-1}^{sim}, x_{t-2}^{sim}, x_{t-3}^{sim}, \hat{\boldsymbol{\theta}}_n), n = 1, 2, \ldots, N = 70. \tag{4.48}$$

It is then relatively straightforward to produce a set of one-step ahead forecasts for the out-of-sample, rebased log-prices,

$$[\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{\tilde{T}}] = \left[0, \log \frac{\tilde{P}_2}{\tilde{P}_1}, \ldots, \log \frac{\tilde{P}_{\tilde{T}}}{\tilde{P}_1}\right], \tag{4.49}$$

where $\tilde{P}_t$ is the observed index value at time t and we consider the period beginning on 02-01-2006 and ending on 02-01-2007. In more detail, assuming we wish to predict $\tilde{x}_{t+1}$ using the information available at time t and further assuming that the model is not refit after each prediction[70], we consider forecasts of the form

$$\hat{x}_{t+1}(\hat{\boldsymbol{\theta}}_n) = \arg\max_x \tilde{p}(x|\tilde{x}_t, \tilde{x}_{t-1}, \tilde{x}_{t-2}, \hat{\boldsymbol{\theta}}_n), n = 1, 2, \ldots, N. \tag{4.50}$$

In other words, the forecast for the value of $\tilde{x}_{t+1}$ at time t associated with parameter set $\hat{\boldsymbol{\theta}}_n$ is simply the point, x, with the maximum[71] estimated density conditional on the realised,

[68] As before, we generate 10 MCMC chains and keep the one associated with the lowest DIC value.

[69] Note that this simply corresponds to the cumulative sum of the log-return process.

[70] Since refitting the model can potentially be quite computationally-expensive for large-scale ABMs, this is likely to often be the case in practice.

[71] Any optimisation routine could be considered, but we opted for a sequential least squares programming (SLSQP) approach, as implemented in Python's SciPy module.

out-of-sample observations at times $t-2$, $t-1$, and t. It then follows that our final forecast is simply given by

$$\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{X})}\left[\hat{x}_{t+1}(\boldsymbol{\theta})\right] = \int \hat{x}_{t+1}(\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{X}) d\boldsymbol{\theta}, \tag{4.51}$$

which we approximate according to

$$\frac{1}{N}\sum_{n=1}^{N} \hat{x}_{t+1}(\hat{\boldsymbol{\theta}}_n). \tag{4.52}$$

Of course, a two-step ahead forecast could also easily be obtained using the recursive forecasting technique, where we first perform a one-step ahead forecast and proceed to repeat the process while replacing $\tilde{p}(x|\tilde{x}_t, \tilde{x}_{t-1}, \tilde{x}_{t-2}, \hat{\boldsymbol{\theta}}_n)$ in Eqn. 4.50 with $\tilde{p}(x|\hat{x}_{t+1}(\hat{\boldsymbol{\theta}}_n), \tilde{x}_t, \tilde{x}_{t-1}, \hat{\boldsymbol{\theta}}_n)$. An analogous procedure also applies for a greater number of prediction steps.

Table 4.7: Predictive Accuracy Summary for Various S&P 500 Log-price Forecasts

	FW-HPM	ARMA-GARCH
DIC	$\mathbf{-32793.8008}$	-32566.4825
MAE, 1-Step Ahead, Bayes	**0.0058**	0.0117
MAE, 2-Step Ahead, Bayes	**0.0078**	0.0151
MAE, 3-Step Ahead, Bayes	**0.0098**	0.0189
MAE, 1-Step Ahead, MAP	**0.0070**	0.0173
MAE, 2-Step Ahead, MAP	**0.0095**	0.0214
MAE, 3-Step Ahead, MAP	**0.0123**	0.0258

The best performing model is indicated in bold. MAE refers to the mean absolute error.

Now, referring to Figure 4.4, which presents the one-, two- and three-step ahead forecasts resulting from the application of the above protocol to both the FW-HPM and ARMA-GARCH models, we observe a key trend worth highlighting. To elaborate, we see that the forecasts produced by the FW-HPM model appear to be much closer to the realised index values than those associated with the ARMA-GARCH model, regardless of the number of forecasting steps[72]. This reinforces the findings of Section 4.4.1.2, with the FW-HPM model again being shown to outperform the benchmark in predictive tasks. A key point worth noting, however, is that in this case we are directly comparing actual forecasts as opposed to constructing likelihood-based approximations and this exercise therefore provides further support for the model selection protocol introduced in the remainder of this chapter.

The above findings are presented in more detail in Table 4.7, which presents the mean absolute errors (MAE) associated with each forecast. Notice that we also distinguish between two estimates for each case, Bayes and MAP, referring to the forecasts associated with integration over the posterior and evaluation at the posterior mode (as proxied by the highest likelihood sample) respectively. As should be clear from the aforementioned table, integration over the posterior noticeably improves the MAE in all cases, suggesting that our modifications to the approach of Delli Gatti and Grazzini (2020) result in meaningful improvements in forecasting performance.

[72]This becomes particularly apparent towards the end of the considered period.

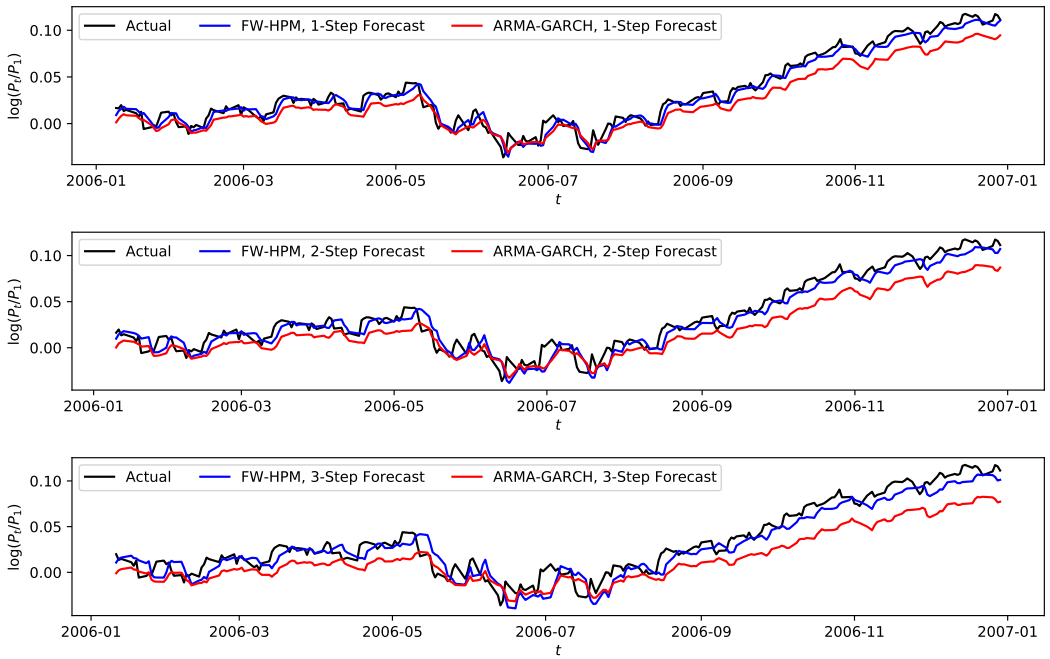


Figure 4.4: One-, two-, and three-step ahead forecasts associated with the FW-HPM and ARMA-GARCH models for the S&P 500 from 02-01-2006 to 02-01-2007.

The above exercises, as well as those included in the remainder of this chapter, follow the tradition in the model selection literature of comparing the predictive accuracy associated with alternative model specifications, such as ABMs, to ARMA-style models of returns that incorporate GARCH or variations thereof to represent the dynamics that govern the time-varying volatility process (Barde 2016; Bertschinger and Mozzhorin 2020). Our results indicate that ABMs, such as the FW-HPM specification, produce return time series dynamics closer to those observed empirically than variations of the ARMA framework incorporating GARCH-style errors, which in turn leads to the greater predictive accuracy associated with ABMs. Of course, what must be clarified, is that while returns can be used as above to compare models, this does not necessarily imply that ABMs can easily forecast returns or prices in the absolute sense with a level of accuracy that could, for example, be traded upon. In general, prices and returns are notoriously difficult to forecast with any reasonable precision and for this reason our inclusion of directional log-price (or cumulative return) forecasting merely acts as a verification of our previous model selection work. It is worth noting, however, that the ideas introduced in this appendix could easily be extended and applied to more tractable problems, such as volatility forecasting, in a dedicated piece of future work.

Chapter 5

Gaussian Process-emulated Adaptive Markov Chain Monte Carlo for Economic Simulation Models

In Chapters 3 and 4, we introduced and extensively tested a neural network-based approach to Bayesian inference for economic ABMs. In particular, we showed that the methodology can indeed be applied to relatively complex and computationally-expensive examples, including the INET Oxford housing market model and a variety of LOB simulators, while still yielding meaningful insights. As previously discussed, this is made possible due to the fact that the methodology attempts to keep the quantity of simulated data required for each likelihood evaluation as small as possible. Nevertheless, and even if the individual likelihood evaluations are tractable, the MCMC samplers required to perform statistical inference are sequential in nature, i.e. difficult to parallelise, and often involve thousands of calls to the likelihood approximation routine. In some cases, this may prove to be a non-trivial exercise and it would therefore be worthwhile to seek ways to further reduce the associated computational costs.

Along these lines, we now demonstrate how recent advances in the surrogate modelling literature may be combined with our neural network-based likelihood approximation methodology and the sampling approach introduced by Griffin and Walker (2013) to yield an emulated, adaptive Metropolis-Hastings algorithm that requires significantly fewer likelihood evaluations in order to produce an equivalent number of posterior samples, while still maintaining the overall quality of the obtained results.

5.1 Surrogate Modelling

From the preceding chapters, it should be clear that regardless of the specific methodology employed, inference for economic ABMs typically involves a common sequence of steps. To be more precise, we generate an ensemble of Monte Carlo replications, $\boldsymbol{X}^{sim}(\boldsymbol{\theta})$, for each considered parameter combination, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, and construct a univariate, real-valued function,

$f(\boldsymbol{\theta})$, that quantifies the similarity or dissimilarity between $\boldsymbol{X}^{sim}(\boldsymbol{\theta})$ and some empirically-measured equivalent. Regardless of whether a Bayesian or frequentist paradigm is adopted, f would need to be evaluated many[1] times during the execution of either MCMC or heuristic optimisation algorithms respectively. During each of these evaluations, new simulated data must be generated, leading to an exercise that may quickly become computationally-intractable as the scale of the considered models is increased.

One possible solution to the above is the use of what is called surrogate or metamodelling. In the economic ABM context, this usually involves the sampling of a relatively small set of initial parameter combinations[2] and the evaluation of the corresponding values of f. Since f is univariate[3], the obtained parameter samples and objective function values can be used as training data for an appropriate regression model, $\hat{f}$, that maps values of $\boldsymbol{\theta}$ to values of f. In principle, the number of evaluations of f needed to construct the regression or surrogate model should be negligible[4] relative to that required in a full MCMC or heuristic optimisation exercise and $\hat{f}$ should also have a negligible cost of evaluation once trained. The original MCMC or heuristic optimisation procedures can then be performed with $\hat{f}$ taking the place of f, shifting much of the computational burden of the inference procedure to the more computationally-tractable surrogate model training step and reducing the overall amount of simulated data that must be generated.

While any regression model could be considered as a surrogate, two major approaches seem to have become dominant in the economic ABM literature, namely Gaussian processes[5] (GPs), which are considered in the work of Salle and Yildizoglu (2014) and Barde and van der Hoog (2017), and gradient boosted trees, which are employed by Lamperti et al. (2018) and Schwartz (2021). Much like the estimation and calibration literature in general, these exercises are primarily frequentist in nature and use surrogate modelling to decrease the computational burden associated with the optimisation of SMD objective functions.

5.1.1 Weaknesses of Global Surrogate Modelling

Surrogate modelling, as it is described above, while appealing in theory, presents non-trivial challenges in practice. In more detail, these approaches essentially construct what are perhaps more precisely described as global surrogates. 'Global', in this case, refers to the fact that these approaches rely on training surrogates that are accurate throughout the parameter space, or 'globally', such that they may act as robust and general replacements for the original functions they aim to emulate. This is a lofty goal, and one that is often not achievable due to a number of important considerations.

Firstly, and as was explored in detail in Chapter 2, most methods for constructing f result in hypersurfaces that are characterised by pronounced slopes in regions of the parameter space that are near the boundaries or relatively distant from local optima. As local optima are approached, however, f begins to flatten and demonstrates less dramatic changes in value for equivalent shifts in $\boldsymbol{\theta}$. Accurately emulating the behaviours observed in these distinct regimes requires a sufficiently flexible model, which, in turn, requires a sufficiently large and

[1] Typically several thousand.

[2] A space-filling design, such as a Latin hypercube or Sobol sequence, is usually considered in this step.

[3] The regression problem is computationally-tractable precisely because f is univariate as opposed to being a high-dimensional time series.

[4] Typically several hundred.

[5] Beyond the economic ABM context, GPs and surrogate modelling have long played an important role in the general engineering literature. We therefore suggest that the interested reader refer to Gramacy (2020) for a more general discussion of both topics.

well-dispersed training set that effectively covers the parameter space, particularly in areas characterised by a flatter hypersurface. This is difficult to achieve, particularly in higher dimensions, where obtaining a representative sample of the parameter space using a limited number of evaluations becomes impractical, even when using a space-filling design. As a result, there is a significant risk that the surrogate model may only manage to capture the most pronounced changes of the objective function hypersurface, while effectively ignoring nuanced changes near local optima. As was again demonstrated in Chapter 2, distortions of the objective function hypersurfaces in these flat regions can dramatically alter the final parameter estimates and undermine the overall performance of the estimation or calibration methodology.

This problem is also further exacerbated by the fact that f may be undefined for certain regions of the parameter space as a result of simulation crashes. These areas can effectively be ignored in most MCMC or heuristic optimisation routines by assigning an arbitrarily poor objective function value whenever the evaluation of f fails. In the surrogate modelling case, however, the assignment of such values may distort the entire surrogate model, especially when the metamodelling approach attempts to model the observation noise. Alternatively, if these problematic points were to simply be excluded, the surrogate model would essentially be performing a form of extrapolation in these areas, which may, depending on the shape of the overall hypersurface, result in them being assigned more favourable objective function values than stable regions of the parameter space and thus draw the MCMC or optimisation algorithms into these undefined regions.

In addition to the above, an equally-important concern relates to the overall accuracy of the surrogate model, which, in most approaches, is unknown. For this reason, it can be difficult to assess the extent to which the consideration of a surrogate has affected the final estimation results. In the studies[6] highlighted in Section 5.1 (notably Barde and van der Hoog (2017)), the quality of the surrogate model is assessed once the heuristic optimisation procedure has concluded by evaluating the objective function at the obtained minimum and comparing it to the surrogate-predicted objective function value. While a procedure of this nature certainly would act as a basic sanity check, it does not account for the fact that the surrogate may be inaccurate for other regions of the parameter space that could have, in fact, contained the true global minimum. Additionally, quantifying the significance of the error between the actual and surrogate objective functions is non-trivial and even differences that appear small to the eye may in fact be significant in regions of the parameter space characterised by a flat hypersurface.

Along these lines, we refer to the recent work of Järvenpää and Corander (2021), who make similar observations and introduce a novel approach that constructs a local GP surrogate in a Bayesian context that we ultimately believe addresses all of the above concerns in a systematic and transparent way. For this reason, the majority of this chapter will focus on discussions of exactly how this is achieved and how their methodology may be adapted to be incorporated into our existing suite of Bayesian inference tools.

5.1.2 Gaussian Processes

Before proceeding with a discussion on the exact approach adopted by Järvenpää and Corander (2021), a brief overview of GPs is presented for the sake of completeness[7].

[6] It is also worth mentioning that most of the aforementioned investigations do not conduct non-emulated experiments against which to compare the final emulated results.

[7] Our discussions borrow heavily from Chapter 15 of Murphy (2012) and Chapter 2 of Rasmussen and Williams (2005).

5.1.2.1 Posteriors Over Functions

In essence, a GP may be seen as a Bayesian approach to regression that begins with the specification of a prior over possible functions, which is then updated in response to a set of training data in order to obtain a corresponding posterior. The specification of distributions over functions is made possible by assuming that the function values, $\boldsymbol{f} = \{f(\boldsymbol{\theta}_1), f(\boldsymbol{\theta}_2), \dots, f(\boldsymbol{\theta}_N)\}$, at a given arbitrary, but finite set of points, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_N\}$, are jointly normally distributed with mean $\boldsymbol{\mu}(\boldsymbol{\theta})$ and covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. In this case, $\Sigma_{i,j} = \kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, where κ is a positive-definite kernel function defining the dissimilarity between $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$.

With reference to the above, we denote the prior over f by

$$f(\boldsymbol{\theta}) \sim GP(m(\boldsymbol{\theta}), \kappa(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{5.1}$$

implying that for any finite set of points, $\boldsymbol{\theta}$,

$$p(\boldsymbol{f}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{K}), \tag{5.2}$$

where $K_{i,j} = \kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ and[8] $\boldsymbol{\mu} = (m(\boldsymbol{\theta}_1), m(\boldsymbol{\theta}_2), \dots, m(\boldsymbol{\theta}_N))$.

Now, assuming a training set of the form $\{(\boldsymbol{\theta}_i, y_i), i = 1, 2, \dots, N\}$, where[9] $y_i = f_i + z$ and $z \sim \mathcal{N}(0, \sigma_y^2)$, and out-of-sample points $\boldsymbol{\theta}^* = \{\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \dots, \boldsymbol{\theta}_{N^*}^*\}$, the GP implies a joint distribution of the form[10]

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{f}^* \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} \boldsymbol{K} + \sigma_y^2 I & \boldsymbol{K}_* \\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**} \end{pmatrix}\right), \tag{5.3}$$

where $K_{**}^{i,j} = \kappa(\boldsymbol{\theta}_i^*, \boldsymbol{\theta}_j^*)$ and $K_*^{i,j} = \kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j^*)$. Standard rules for the conditioning of multivariate Gaussian random variables apply, yielding a posterior given by

$$p(\boldsymbol{f}^*|\boldsymbol{\theta}^*, \boldsymbol{\theta}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{f}^*|\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*), \tag{5.4}$$

where

$$\boldsymbol{\mu}^* = \boldsymbol{K}_*^T \boldsymbol{K}_y^{-1} \boldsymbol{y}, \tag{5.5}$$

$$\boldsymbol{\Sigma}^* = \boldsymbol{K}_{**} - \boldsymbol{K}_*^T \boldsymbol{K}_y^{-1} \boldsymbol{K}_*, \tag{5.6}$$

$\boldsymbol{K}_y = \boldsymbol{K} + \sigma_y^2 I$ and I is an $N \times N$ identity matrix.

From the above, two key advantages of GPs should be apparent. Firstly, by virtue of being Bayesian, they provide not only point estimates (the posterior means), but also associated measures of uncertainty (the posterior standard deviations). Additionally, the aforementioned posterior is available in closed-form and Gaussian in nature, implying that the behaviour of the trained regression model is amenable to a detailed theoretical analysis (Murphy 2012; Rasmussen and Williams 2005). Both features are of critical importance to the surrogate modelling scheme proposed by Järvenpää and Corander (2021), making GPs a natural choice.

[8] $m(\boldsymbol{\theta})$ is often simply set to 0.

[9] Since $f(\boldsymbol{\theta})$ is usually estimated from a finite set of Monte Carlo replications, $\boldsymbol{X}^{sim}(\boldsymbol{\theta})$, we must account for some degree of observation noise.

[10] As is standard in the literature (Murphy 2012), we have assumed a zero mean prior.

5.1.2.2 Kernels and Hyperparameter Optimisation

In general, the types of functions that can be modelled by a GP depend strongly on the chosen kernel function, κ. Perhaps the most common choice in practice is the squared exponential or radial basis function (RBF) kernel, which assumes that f is infinitely differentiable and is given by

$$\kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \sigma_f^2 \exp(-\frac{1}{2}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)^T diag(\boldsymbol{l})^{-2}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)), \tag{5.7}$$

where σ_f and $\boldsymbol{l} = (l_1, l_2, \ldots, l_d)$ are hyperparameters and d is the dimensionality of $\boldsymbol{\theta}_i$. These hyperparameters are usually determined via the maximisation of the log marginal likelihood on the training data,

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^T \boldsymbol{K}_y^{-1}\boldsymbol{y} - \frac{1}{2} log|\boldsymbol{K}_y| - \frac{N}{2}\log 2\pi. \tag{5.8}$$

Although the RBF kernel is a common choice, the assumption of infinite differentiability may, in some cases, be too restrictive. Therefore, a more flexible alternative also worth considering is the Matern kernel, given by

$$\kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} d\right)^{\nu} K_\nu \left(\sqrt{2\nu} d\right), \tag{5.9}$$

where $d = ||diag(\boldsymbol{l})^{-1}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)||_2$, Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and ν is a hyperparameter[11].

5.2 The Approach of Järvenpää and Corander (2021)

Following on from the above overview of surrogate modelling, a more detailed discussion of our chosen approach is now in order. In essence, Järvenpää and Corander (2021) begin by constructing a GP surrogate for the log-likelihood based on an initial set of noisy, simulation-based likelihood evaluations, much like standard global surrogate methods. In contrast to these approaches, however, additional rounds of surrogate training are also embedded in the MCMC algorithm that attempts to sample the simulation model parameter posterior.

To elaborate, the error associated with the decision to accept or reject a particular model parameter combination (as a result of using a GP surrogate rather than the actual simulation-based likelihood) is calculated during each iteration of the MCMC algorithm and then compared to a user-specified threshold. If the calculated error value falls below this threshold, the MCMC algorithm proceeds as normal using the surrogate likelihood. If not, new points are systematically added to the training data and the GP retrained until the calculated uncertainty reaches a suitable level.

This ensures that the surrogate likelihood is employed only when it is suitably accurate and, in practice, results in most of the likelihood evaluation budget being used to improve the surrogate in regions of the parameter space that are actually explored by the MCMC algorithm. As a result, the procedure essentially makes use of a local rather than global likelihood surrogate.

[11] The resultant Gaussian process is $\lceil \nu \rceil - 1$ times differentiable.

5.2.1 The Surrogate Log-likelihood

As previously stated, Järvenpää and Corander (2021) iteratively construct a GP surrogate for the log-likelihood function. This involves the generation of a training set, $\mathcal{D}_j = \{(\boldsymbol{\theta}_i, y_i), i = 1, 2, \ldots, N_j\}, j = 1, 2, \ldots, S$, where S is the desired number of iterations of the MCMC algorithm and the index j indicates that the training set evolves during each of these iterations. As in Section 5.1.2, $y_i = f(\boldsymbol{\theta}_i) + z$, where $z \sim \mathcal{N}(0, \sigma_y^2)$ is an observation-level noise term, and y_i is a simulation-based log-likelihood approximator evaluated at $\boldsymbol{\theta}_i$.

In the original investigation conducted by Järvenpää and Corander (2021), y_i is obtained using a relatively simple, summary statistic-based approach known as synthetic likelihood[12] (Wood 2010) and σ_y^2, the observation-level noise variance, is assumed to vary from observation to observation. In our case, however, y_i is obtained using the more sophisticated, summary statistic-free methodology developed in Chapter 3 and we additionally assume a constant noise variance across all of the observations[13]. Nevertheless, the surrogate training process and the resultant equations remain unchanged.

We begin by first assuming a hierarchical[14] GP prior of the form

$$f(\boldsymbol{\theta})|\boldsymbol{\beta} \sim GP(m(\boldsymbol{\theta}), \kappa(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{5.10}$$

where[15]

$$m(\boldsymbol{\theta}) = \sum_{i=1}^{q} \beta_i h_i(\boldsymbol{\theta}) \text{ and } \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B}). \tag{5.11}$$

As suggested by Järvenpää and Corander (2021), the hierarchical Bayesian hyperparameters, $\boldsymbol{\beta}$, are integrated out, leading to the updated GP prior

$$f(\boldsymbol{\theta}) \sim GP(\boldsymbol{h}(\boldsymbol{\theta})^T \boldsymbol{b}, \kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') + \boldsymbol{h}(\boldsymbol{\theta})^T \boldsymbol{B} \boldsymbol{h}(\boldsymbol{\theta}')), \tag{5.12}$$

where $\boldsymbol{h}(\boldsymbol{\theta})$ is a column vector in $\mathbb{R}^q$ with element i being equal to $h_i(\boldsymbol{\theta})$.

Assuming that the above prior is vague, i.e. $\boldsymbol{b} = 0$ and $\boldsymbol{B}^{-1} \to \boldsymbol{0}$, where $\boldsymbol{0}$ is the zero matrix, we have a posterior of the form (Rasmussen and Williams 2005)

$$p(\boldsymbol{f}^*|\boldsymbol{\theta}^*, \mathcal{D}_j) = \mathcal{N}(\boldsymbol{f}^*|\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*), \tag{5.13}$$

where

$$\boldsymbol{\mu}^* = \boldsymbol{K}_*^T \boldsymbol{K}_y^{-1} \boldsymbol{y} + \boldsymbol{R}^T \bar{\boldsymbol{\beta}}, \tag{5.14}$$

$$\boldsymbol{\Sigma}^* = \boldsymbol{K}_{**} - \boldsymbol{K}_*^T \boldsymbol{K}_y^{-1} \boldsymbol{K}_* + \boldsymbol{R}^T (\boldsymbol{H} \boldsymbol{K}_y^{-1} \boldsymbol{H}^T)^{-1} \boldsymbol{R}. \tag{5.15}$$

In the above, $\bar{\boldsymbol{\beta}} = (\boldsymbol{H} \boldsymbol{K}_y^{-1} \boldsymbol{H}^T)^{-1} \boldsymbol{H} \boldsymbol{K}_y^{-1} \boldsymbol{y}$, $\boldsymbol{R} = \boldsymbol{H}_* - \boldsymbol{H} \boldsymbol{K}_y^{-1} \boldsymbol{K}_*$, and $\boldsymbol{H}$ is the matrix whose columns are equal to the $\boldsymbol{h}$ vectors associated with $\boldsymbol{\theta}$, with $\boldsymbol{H}_*$ being similarly defined for $\boldsymbol{\theta}^*$.

[12] Synthetic likelihood assumes that the joint distribution of the summary statistics is multivariate normal, leading to a simple expression for the likelihood. The interested reader should refer to Wood (2010) and Price et al. (2018) for more detailed discussions.

[13] We discuss the implications of this and other modifications made in Section 5.2.4.

[14] See Chapter 5 of Gelman et al. (2004) for a detailed introduction to hierarchical Bayesian modelling.

[15] We assume $h_1(\boldsymbol{\theta}) = d$, $h_2(\boldsymbol{\theta}) = \sum_{k=1}^{d} \theta_k$, and $h_3(\boldsymbol{\theta}) = \sum_{k=1}^{d} \theta_k^2$, where θ_k is the k-th element and d the dimensionality of $\boldsymbol{\theta}$.

Finally, in order to determine suitable hyperparameters for the above GP, $\boldsymbol{\phi}$, we follow the standard procedure of maximising the log marginal likelihood, which, in this context, is given by

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^T\boldsymbol{K}_y^{-1}\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}\boldsymbol{y} - \frac{1}{2}log|\boldsymbol{K}_y| - \frac{1}{2}log|\boldsymbol{A}| - \frac{N-M}{2}\log 2\pi, \tag{5.16}$$

where $\boldsymbol{A} = \boldsymbol{H}\boldsymbol{K}_y^{-1}\boldsymbol{H}^T$, $\boldsymbol{C} = \boldsymbol{K}_y^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{K}_y^{-1}$, and M is the rank of $\boldsymbol{H}^T$.

5.2.2 The Metropolis-Hastings Acceptance Decision

The use of the surrogate log-likelihood constructed in Section 5.2.1 as opposed to actual log-likelihood evaluations introduces a degree of uncertainty into the acceptance decision of the Metropolis-Hastings algorithm. In some cases, this uncertainty may be relatively large and the use of the surrogate may lead to an incorrect decision to accept (reject) a candidate set of parameter values. We therefore require an estimate of the introduced error such that we might be able to identify when the use of the GP surrogate is likely to influence the above decision[16].

To begin, recall that the acceptance probability is usually given by

$$\begin{aligned} \alpha &= \min\left(1, \gamma(\boldsymbol{\theta}, \boldsymbol{\theta}')\right), \\ \gamma(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \frac{\exp(f(\boldsymbol{\theta}'))p(\boldsymbol{\theta}')q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{\exp(f(\boldsymbol{\theta}))p(\boldsymbol{\theta})q(\boldsymbol{\theta}'|\boldsymbol{\theta})}, \end{aligned} \tag{5.17}$$

where $\boldsymbol{\theta}$ is the last accepted parameter set, $\boldsymbol{\theta}'$ is a candidate parameter set (to be accepted or rejected), $f(\boldsymbol{\theta})$ is the log-likelihood, $p(\boldsymbol{\theta})$ is the prior, and $q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ is the proposal distribution. Assuming that the above decision is to be made at iteration j and that the trained log-likelihood surrogate follows the GP specification presented in Section 5.2.1, we have

$$\begin{aligned} \begin{pmatrix} f(\boldsymbol{\theta}) \\ f(\boldsymbol{\theta}') \end{pmatrix} |\mathcal{D}_j &\sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \\ &= \mathcal{N}\left(\begin{pmatrix} m_j(\boldsymbol{\theta}) \\ m_j(\boldsymbol{\theta}') \end{pmatrix}, \begin{pmatrix} s_j^2(\boldsymbol{\theta}) & c_j(\boldsymbol{\theta}, \boldsymbol{\theta}') \\ c_j(\boldsymbol{\theta}', \boldsymbol{\theta}) & s_j^2(\boldsymbol{\theta}') \end{pmatrix}\right). \end{aligned} \tag{5.18}$$

It thus follows that

$$\gamma(\boldsymbol{\theta}, \boldsymbol{\theta}')|\mathcal{D}_j \sim \exp\left[\mathcal{N}(\mu_j(\boldsymbol{\theta}, \boldsymbol{\theta}'), \sigma_j^2(\boldsymbol{\theta}, \boldsymbol{\theta}'))\right], \tag{5.19}$$

when employing the GP surrogate, where

$$\begin{aligned} \mu_j(\boldsymbol{\theta}, \boldsymbol{\theta}') &= m_j(\boldsymbol{\theta}') - m_j(\boldsymbol{\theta}) + \log\frac{p(\boldsymbol{\theta}')q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{p(\boldsymbol{\theta})q(\boldsymbol{\theta}'|\boldsymbol{\theta})}, \\ \sigma_j^2(\boldsymbol{\theta}, \boldsymbol{\theta}') &= s_j^2(\boldsymbol{\theta}') + s_j^2(\boldsymbol{\theta}) - 2c_j(\boldsymbol{\theta}, \boldsymbol{\theta}'). \end{aligned} \tag{5.20}$$

Applying a Bayesian decision-theoretic analysis[17] to the distribution of $\gamma(\boldsymbol{\theta}, \boldsymbol{\theta}')$ obtained above, Järvenpää and Corander (2021) show that the optimal point estimate is

$$\hat{\gamma}_j = e^{\mu_j(\boldsymbol{\theta}, \boldsymbol{\theta}')}, \tag{5.21}$$

[16] Järvenpää and Corander (2021) make use of a single-sample Metropolis-Hastings algorithm, whereas we make use of the sample set-based approach of Griffin and Walker (2013). It should be noted, however, that the determination of the Metropolis-Hastings decision error remains the same in either case. See Section 5.2.4 for more details.

[17] This analysis and the related derivation of the Metropolis-Hastings decision error is fairly technical and beyond the scope of this thesis. The interested reader should, however, refer to Appendix A of Järvenpää and Corander (2021) for full details.

or, in other words, the GP posterior median. Further, they additionally show that the associated Metropolis-Hastings decision error, i.e. the probability of rejecting a parameter set when it should be accepted or accepting it when it should be rejected, is given by

$$E_{j,\hat{\gamma}_j}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \begin{cases} \Phi(-\mu_j/\sigma_j) - e^{\mu_j + \frac{\sigma_j^2}{2}} \Phi(-(\mu_j + \sigma_j^2)/\sigma_j) & \text{if } \mu_j \geq 0, \\ \Phi(\mu_j/\sigma_j) + e^{\mu_j + \frac{\sigma_j^2}{2}} \left[\Phi(-(\mu_j + \sigma_j^2)/\sigma_j) - 2\Phi(-\sigma_j)\right] & \text{if } \mu_j < 0, \end{cases} \tag{5.22}$$

where $\mu_j = \mu_j(\boldsymbol{\theta}, \boldsymbol{\theta}')$, $\sigma_j = \sigma_j(\boldsymbol{\theta}, \boldsymbol{\theta}')$, and Φ is the standard normal cumulative density function.

5.2.3 Retraining the Surrogate

With an estimate of the Metropolis-Hastings decision error now readily available, it is relatively straightforward to determine if, during a given iteration j, the use of the GP surrogate is likely to distort the Metropolis-Hastings acceptance decision. We achieve this by comparing $E_{j,\hat{\gamma}}$ to a user-specified tolerance parameter, ϵ. If $E_{j,\hat{\gamma}} \leq \epsilon$, we base the acceptance decision on $\hat{\gamma}_j$ as is. If $E_{j,\hat{\gamma}} > \epsilon$, however, we add new parameter samples and corresponding log-likelihood evaluations to $\mathcal{D}_j$ and retrain the GP until $E_{j,\hat{\gamma}}$ falls below the desired tolerance.

Of course, the addition of randomly-sampled parameter combinations during this process would be highly-inefficient and in many cases produce little to no improvement in the decision error. A preferable alternative is the determination of parameter combinations that result in the greatest possible error reduction at each step, $\boldsymbol{\theta}_{opt}$. Järvenpää and Corander (2021) show that such combinations may be systematically determined by setting

$$\boldsymbol{\theta}_{opt} = \arg\max_{\boldsymbol{\theta}^* \in \boldsymbol{\Theta}} [c_j(\boldsymbol{\theta}, \boldsymbol{\theta}^*) - c_j(\boldsymbol{\theta}', \boldsymbol{\theta}^*)][c_j(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) + \sigma_y^2]^{-1}[c_j(\boldsymbol{\theta}^*, \boldsymbol{\theta}) - c_j(\boldsymbol{\theta}^*, \boldsymbol{\theta}')]. \tag{5.23}$$

5.2.4 Full Algorithm and Modifications

Combining the components introduced above, we obtain the complete GP-emulated adaptive Metropolis-Hastings procedure, which we outline in Algorithm 1.

Although we follow the specification provided by Järvenpää and Corander (2021) for the most part, it should be noted that we have modified aspects of the methodology. Firstly, instead of using a traditional, single-sample Metropolis-Hastings algorithm, we make use of the sample set-based approach of Griffin and Walker (2013), which has been our method of choice for MCMC throughout this thesis. In contrast to the original work of Järvenpää and Corander (2021), who find single-sample approaches to be quite sensitive to the location of the initial sample, we find the emulated Griffin and Walker (2013) algorithm to be more or less robust to the choice of initial sample set. Therefore, in this respect, our work may be seen as an iterative improvement to the original procedure.

Additionally, and as previously mentioned, we have also replaced their employed summary statistic-based likelihood construction (synthetic likelihood) with our own summary statistic-free alternative. Although the original GP-emulated MCMC methodology is fairly general and designed for arbitrary simulation-based likelihood approximations, a minor adjustment was required for compatibility with our neural network-based approach. To be more precise, Järvenpää and Corander (2021) consider a variable observation-level noise intensity, σ_y^i, that is calculated separately for each synthetic likelihood evaluation, i. This is

Algorithm 1 GP-emulated Adaptive Metropolis-Hastings

Input: Prior $p(\boldsymbol{\theta})$, procedure for computing noisy log-likelihoods y_i, GP prior for f, error tolerance ϵ, parameter ranges to be explored $\boldsymbol{\Theta}_{init}$, initial training set size N_0, number of sample sets S, number of samples per sample set N_{set}

Output: S approximate Griffin and Walker (2013) sample sets of size N_{set}

Generate a set of initial parameter combinations, $\boldsymbol{\theta}_i, i = 1, 2, \ldots, N_0$, which are randomly drawn from $\boldsymbol{\Theta}_{init} \subseteq \boldsymbol{\Theta}$ ▷ Use Latin hypercube sampling
for $i = 1 : N_0$ **do**
    Evaluate y_i at $\boldsymbol{\theta}_i$ ▷ Use the MDN method introduced in Chapter 3
end for
Create the initial training set, $\mathcal{D} \leftarrow \{(\boldsymbol{\theta}_i, y_i), i = 1, 2, \ldots, N_0\}$
Fit the GP surrogate and optimise the kernel hyperparameters using $\mathcal{D}$ ▷ We also treat σ_y as a hyperparameter to be optimised during this step
Generate an initial Griffin and Walker (2013) sample set, $\boldsymbol{\theta}_0^{(j)}, j = 1, 2, \ldots, N_{set}$, where each $\boldsymbol{\theta}_0^{(j)}$ is randomly drawn from $\boldsymbol{\Theta}_{init}$ ▷ Use simple random sampling
for s = 1 : S **do**
    Generate a random integer, $k \in \{1, 2, \ldots, N_{set}\}$, and set $\boldsymbol{\theta}_{old} \leftarrow \boldsymbol{\theta}_{s-1}^{(k)}$
    Sample $\boldsymbol{\theta}' \sim q\left(\cdot\middle|\boldsymbol{\theta}_{old}\right) = q\left(\cdot\middle|\boldsymbol{\theta}_{s-1}^{(1)}, \boldsymbol{\theta}_{s-1}^{(2)}, \ldots, \boldsymbol{\theta}_{s-1}^{(N_{set})}\right)$, i.e. from the Griffin and Walker (2013) proposal
    while $E_{\hat{\gamma}}(\boldsymbol{\theta}_{old}, \boldsymbol{\theta}') > \epsilon$ **do** ▷ $q\left(\cdot\middle|\boldsymbol{\theta}'\right) = q\left(\cdot\middle|\boldsymbol{\theta}_{s-1}^{(1)}, \ldots, \boldsymbol{\theta}_{s-1}^{(k-1)}, \boldsymbol{\theta}', \boldsymbol{\theta}_{s-1}^{(k+1)}, \ldots, \boldsymbol{\theta}_{s-1}^{(N_{set})}\right)$
        Solve for $\boldsymbol{\theta}_{opt}$ as described in Eqn. 5.23
        Evaluate y_{opt} at $\boldsymbol{\theta}_{opt}$
        Update the training set, $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{\theta}_{opt}, y_{opt})\}$
        Fit the GP surrogate and optimise the kernel hyperparameters using $\mathcal{D}$
    end while
    Calculate $\hat{\gamma}$ as described in Eqn. 5.21
    Set $\hat{\alpha} \leftarrow \min(1, \hat{\gamma})$
    if $u \leq \hat{\alpha}, u \sim \mathcal{U}(0, 1)$ **then**
        Set $\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N_{set})} \leftarrow \boldsymbol{\theta}_{s-1}^{(1)}, \ldots, \boldsymbol{\theta}_{s-1}^{(k-1)}, \boldsymbol{\theta}', \boldsymbol{\theta}_{s-1}^{(k+1)}, \ldots, \boldsymbol{\theta}_{s-1}^{(N_{set})}$
    else
        Set $\boldsymbol{\theta}_s^{(1)}, \boldsymbol{\theta}_s^{(2)}, \ldots, \boldsymbol{\theta}_s^{(N_{set})} \leftarrow \boldsymbol{\theta}_{s-1}^{(1)}, \boldsymbol{\theta}_{s-1}^{(2)}, \ldots, \boldsymbol{\theta}_{s-1}^{(N_{set})}$
    end if
end for

made possible through a simple bootstrapping procedure that leverages the strong assumptions made by synthetic likelihood.

Unfortunately, our approach does not provide an equivalent procedure. We therefore limit ourselves to an assumption of homogenous observation-level noise and instead treat σ_y as a hyperparameter that is determined via the maximisation of the log marginal likelihood[18]. Nevertheless, as we will soon show, this has little effect on the ability of the GP-emulated MCMC algorithm to effectively sample the posteriors associated with our likelihood construction methodology and therefore seems to be a very minor adjustment.

5.3 Results and Discussion

Having now completely specified the GP-emulated Metropolis-Hastings approach, we proceed with the presentation of our results. Overall, our analysis is primarily focused on determining if, firstly, the emulated MCMC algorithm accurately recovers the posteriors produced by the non-emulated procedure, and, secondly, if the use of emulation results in a meaningful reduction in the computational time required for our considered set of problems.

Naturally, the first of these aspects would require a means to measure the discrepancy between the original and emulated posteriors. Following the recommendation of Järvenpää and Corander (2021), we make use of the total variation (TV) distance, which takes values in the interval $[0, 1]$ and is thus easy to interpret[19]. Additionally, we also require a set of examples for which to perform both of the aforementioned tests. Along these lines, we consider a number of progressively more difficult test cases originally introduced in Chapter 3, including the random walk, Brock and Hommes (1998), and INET Oxford housing market models.

In all of the above cases and unless otherwise stated, we consider an identical experimental setup to that employed in the original experiments, changing only the sampling methodology, which has been replaced with the emulated Metropolis-Hastings algorithm introduced in this chapter. During each run of the emulated sampler, we consider $N_0 = 75$ initial log-likelihood evaluations, a total of $S = 25000$ sample sets[20] (with a burning-in period of 1500 sample sets), and $N_{set} = 70$ samples per sample set[21].

5.3.1 Random Walk Model

To begin our analysis, we proceed by investigating the fourth free parameter set of the random walk model originally considered in Chapter 3, where we attempted to correctly identify the pre- and post-break drift, or d_1 and d_2 respectively. When emulating this experiment, we consider multiple error tolerances, $\epsilon = \{0.25, 0.35\}$, and two different kernels, RBF and Matern with $\nu = 1.5$, in order to provide a basic analysis of the sensitivity of the emulated Metropolis-Hastings algorithm to differences in the chosen hyperparameters. Additionally, for each kernel and error tolerance combination, we consider 10 runs of the

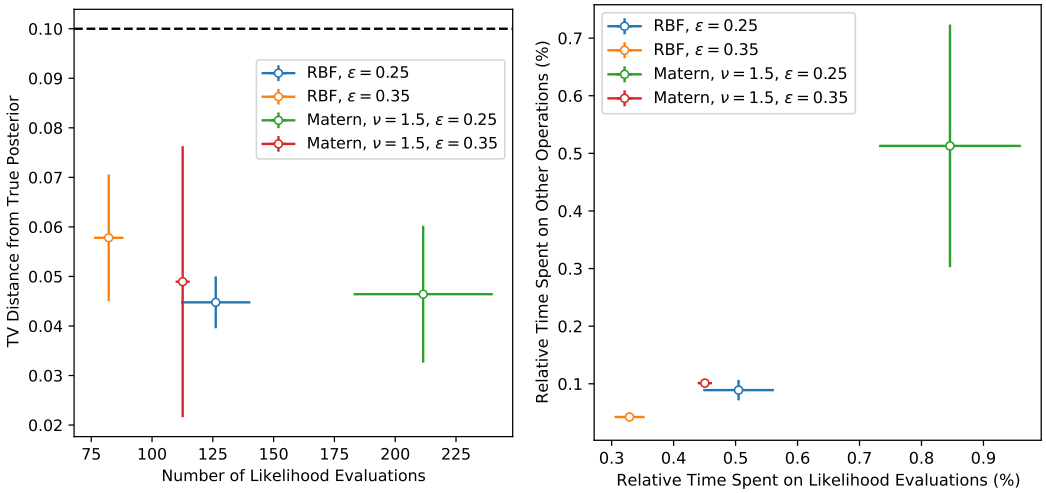
[18] As stated by Rasmussen and Williams (2005), this is, in fact, the standard approach for determining the observation-level noise intensity.

[19] We consider values less than 0.15 to correspond to reasonable approximations. See Appendix 5.A for more details.

[20] This is because we considered 5 separate chains of length 5000 each in the original ground-truth experiments to generate a total of 25000 sample sets.

[21] Further experimental details are provided in Appendix 5.B.

entire emulation process to investigate the effects of different initial conditions on the overall performance of the methodology.



(a) Posterior Approximation Quality

(b) Execution Time

Figure 5.1: Performance metrics for the GP-emulated random walk model estimation experiments. In all cases, we show the mean and one standard deviation error bars over 10 independent runs of the emulated Metropolis-Hastings algorithm, with a different random initial sample set being considered during each run. All execution time measurements are expressed as a percentage of the approximate time required to generate an equivalent number of posterior samples without emulation.

Referring to Figure 5.1a, which presents the TV distances associated with the aforementioned emulation experiments, we notice that, in all cases, the obtained distances are well below our threshold of 0.15 and indeed even the stricter threshold of 0.1. This indicates that, regardless of the chosen hyperparameter settings, the emulated posteriors are high-quality approximations to the original. As expected, we also find that decreasing the error tolerance results in a reduction in the mean TV distance and an increase in the mean number of likelihood evaluations required, indicating that a stricter tolerance leads to higher-quality posterior approximations at the expense of computational-efficiency. Further, it is also worth noting that decreasing the error tolerance results in reduced variation in the obtained TV distances, but increased variation in the number of required likelihood evaluations, which also appeared to be the case in the experiments of Järvenpää and Corander (2021). It therefore seems that the procedure has successfully emulated the random walk model experiment and that the behaviour of the methodology is similar to that observed in the original investigation conducted by Järvenpää and Corander (2021), despite our consideration of a different likelihood construction process and some of the other modifications made.

A more detailed analysis of the experiments involving an RBF kernel and $\epsilon = 0.25$ is presented in Figure 5.2, where we plot the GP-emulated posterior marginals. In line with our interpretation of the obtained TV distances, we observe that the emulated posteriors are well-behaved, comparable from one emulated sampling run to the next, and all peak around the posterior mean obtained in the original, non-emulated experiment.

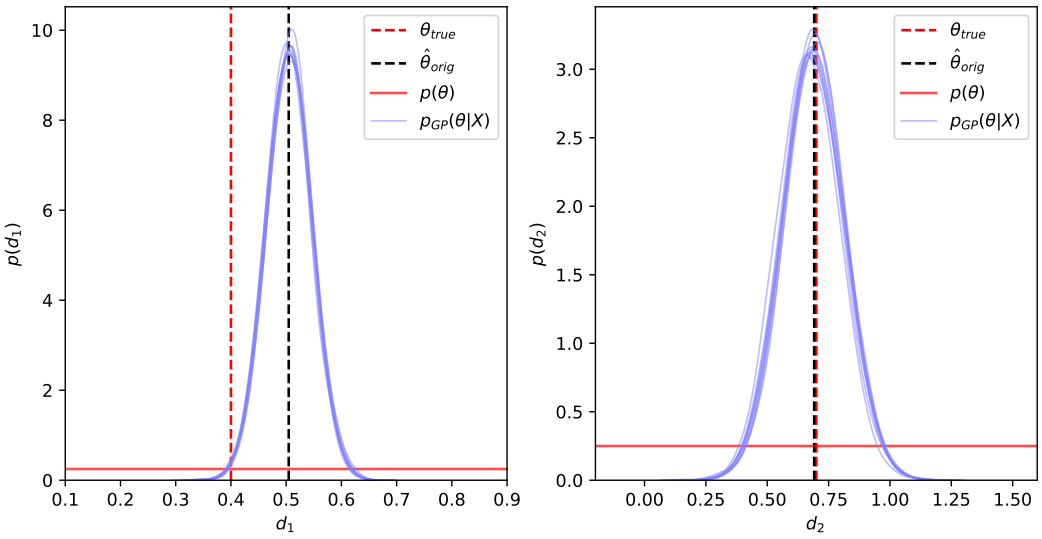


Figure 5.2: Marginal posterior distributions, $p_{GP}(\theta|\boldsymbol{X})$, for the random walk model obtained from 10 independent runs of the GP-emulated Metropolis-Hastings algorithm with an RBF kernel and $\epsilon = 0.25$. $\hat{\theta}_{orig}$, the posterior mean obtained in the original, non-emulated experiments, is plotted for comparison.

Of course, the accuracy of the emulation procedure is only relevant when assessed in light of the overall computational costs. Fortunately, it turns out that the resultant computational savings are substantial. Firstly, as shown in Figure 5.1a, just over 200 likelihood evaluations are required in the worst case, substantially less than the 25000 required to produce the same number of samples without emulation. Additionally, Figure 5.1b, which expresses the computational time required to perform all likelihood evaluations (x-axis) and other tasks such as GP training and optimisation (y-axis), reveals that, in the worst case, the emulated Metropolis-Hastings algorithm utilises just 1.5% of the time required by the non-emulated procedure[22].

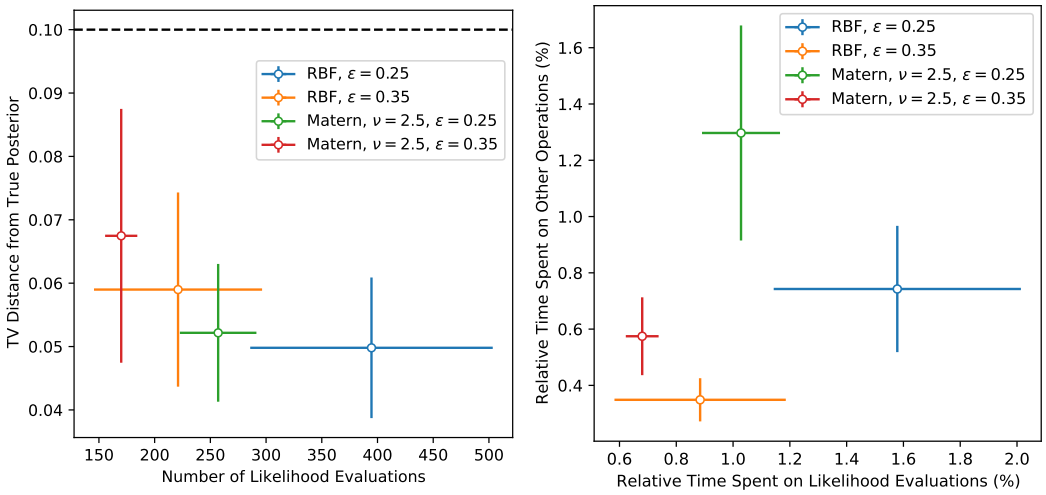
5.3.2 Brock and Hommes (1998) Model

Following our success in the context of the random walk model, a natural next step is the determination of the extent to which similar findings apply for a more complex example. Along these lines, we now consider the first free parameter set of the Brock and Hommes (1998) model, which was also originally presented in Chapter 3 and represents a greater challenge in terms of the complexity of the considered model and the number of free parameters[23]. Referring to Figures 5.3 and 5.4, we see that despite this increased difficulty,

[22]Both axes of Figure 5.1b correspond to percentages of the full, non-emulated experiment time. To summarise, we measure the total computational time used to generate likelihood evaluations (x-axis) and perform all other operations (y-axis) during each emulation run and divide these values by $25000\mu_{time}$, where μ_{time} is the average time taken to perform a single likelihood evaluation over the course of the emulation run. We then multiply the values on each axis by 100 to obtain percentages.

[23]As in the case of our investigation into the random walk model, we replicate the conditions of the original, non-emulated experiment. It should be noted, however, that we have narrowed the ranges associated with $\boldsymbol{\Theta}_{init}$ to $g_2 \in [-1.5, 0]$, $b_2 \in [-0.5, 0]$, $g_3 \in [0, 1.5]$, and $b_3 \in [0, 0.5]$. We have opted to make this slight

identical interpretations apply, with the emulation procedure again producing high-quality approximations to the original posterior at a fraction of the computational cost.



(a) Posterior Approximation Quality

(b) Execution Time

Figure 5.3: Performance metrics for the GP-emulated Brock and Hommes (1998) model estimation experiments. In all cases, we show the mean and one standard deviation error bars over 10 independent runs of the emulated Metropolis-Hastings algorithm, with a different random initial sample set being considered during each run. All execution time measurements are expressed as a percentage of the approximate time required to generate an equivalent number of posterior samples without emulation.

At this point, some particularly observant readers may have noticed that we have not yet directly compared the RBF and Matern kernels in our discussion and that we further consider a Matern kernel with $\nu = 2.5$ in this section as opposed to the original value of $\nu = 1.5$ investigated in the random walk experiments. We justify this as follows. Firstly, when we originally considered $\nu = 1.5$ in our attempts to emulate the Brock and Hommes (1998) model experiment, it was not uncommon to encounter non-convergence. To elaborate, we typically observe that, in successful applications with other kernels, the first few hundred iterations of the adaptive Metropolis-Hastings algorithm are relatively slow as a result of the posterior region having not yet been approached and additional examples being iteratively added to the GP training set in order to keep the decision error below ϵ. After this initial iterative training period, however, a type of convergent behaviour is observed in which the decision error remains below ϵ without any need to retrain the GP[24].

In the case of the original Matern kernel with $\nu = 1.5$, we observed that the hyperparameter tuning process would often set σ_y to its minimum value, suggesting that the GP may

modification primarily because the sampler initialisation ranges that we originally considered were selected to accommodate the fact that the approach of Grazzini et al. (2017) tended to produce much wider posteriors and struggled to effectively narrow down the parameter space to a precise region. As illustrated in Chapter 3, our methodology does not suffer from this problem and we therefore consider a more realistic and less extreme set of initialisation ranges. It should be noted, however, that we do maintain the original priors and that the initialisation ranges are still wide relative to the posteriors.

[24]We discuss this in more detail in Section 5.4.

have been overfitting the training data. This, in turn, seems to have led to non-convergence, with the GP-emulated Metropolis-Hastings algorithm taking a substantial amount of time to reach a point where the retraining of the GP becomes infrequent. In response, we instead considered a higher ν value in order to reduce the overall complexity of the GP model and found this to improve the stability of the method[25].

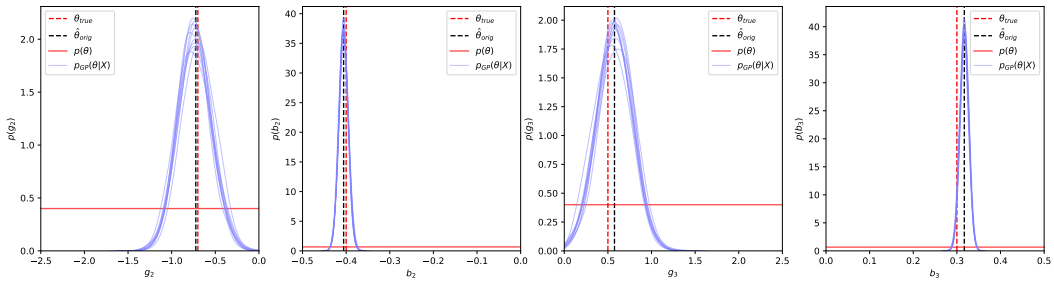


Figure 5.4: Marginal posterior distributions, $p_{GP}(\theta|\boldsymbol{X})$, for the Brock and Hommes (1998) model obtained from 10 independent runs of the GP-emulated Metropolis-Hastings algorithm with an RBF kernel and $\epsilon = 0.25$. $\hat{\theta}_{orig}$, the posterior mean obtained in the original, non-emulated experiments, is plotted for comparison. Also, notice the similarity when compared to Figure 3.1.

Unfortunately, with the exception of the Matern kernel tending to produce a greater range of variation in the obtained TV distances for a given value of ϵ, there do not appear to be any definitive and consistent trends in the performance differences between the RBF and Matern kernels in our experiments. Despite this, and given that the Matern kernel does, at least in some cases, seem to be associated with overfitting, we believe that the RBF kernel may be a preferable starting point for most applications.

5.3.3 INET Oxford Housing Market Model

Following on from our consideration of illustrative examples involving simple models, we now proceed to the investigation of a far more realistic test case: the INET Oxford housing market model. Ultimately, it is models of this class for which GP emulation is likely to be most crucial and it is therefore imperative that the results obtained previously are able to generalise to this more challenging context. We thus attempt to emulate the experiment presented in Section 3.4.2, where we estimated 4 of the model's key parameters.

It should be noted, however, that we do make a slight adjustment. Recall, from Figure 3.7, that the posterior associated with *P Investor* has a relatively long tail. As stated by Järvenpää and Corander (2021), posteriors exhibiting such behaviours may not be well-suited to their approach or emulation in general. We have therefore opted not to consider such a complication at this stage and instead estimate the *Desired Rent Income Fraction* parameter in place of *P Investor*[26].

This does, of course, raise the question of the limitations of the methodology and indeed surrogate models in general. We discuss these and related issues in Section 5.4. Nevertheless,

[25] Recall that higher values of ν correspond to smoother GP function approximations.

[26] All other aspects of the experiment remain unchanged. *Desired Rent Income Fraction* is assumed to lie in the interval $[0, 0.5]$.

the modified example considered in this section is by no means trivial and still provides a challenging test for the protocol and estimation techniques in general.

Table 5.1: Emulation Result Summary for the Housing Market Model (RBF, $\epsilon = 0.35$)

	θ_1	θ_2	θ_3	θ_4
Ground-truth				
$\boldsymbol{\theta}_{true}$	0.25	0.05	0.33	0.5
Non-emulated Experiment				
$\boldsymbol{\mu}_{posterior}$	0.2845	0.0467	0.3030	0.5683
$\boldsymbol{\sigma}_{posterior}$	0.0636	0.0071	0.0267	0.0466
GP-emulated Experiment				
$\boldsymbol{\mu}_{posterior}$	0.2780	0.0466	0.3023	0.5696
	(0.0106)	(0.0010)	(0.0044)	(0.0081)
$\boldsymbol{\sigma}_{posterior}$	0.0655	0.0075	0.0323	0.0483
	(0.0170)	(0.0025)	(0.0184)	(0.0173)
	Mean	Std. Dev.	Median	MAD
Performance Metrics				
TV Distance	0.1231	0.0227	0.1226	0.0160
Likelihood Evals.	838.50	226.81	803.50	146.50
Time Spent on Evals.	3.35%	0.91%	3.21%	0.59%
Time Spent on Other Ops.	7.57%	5.63%	5.93%	3.34%

θ_1-θ_4 correspond to, in order, *Market Average Price Decay*, *Sale Epsilon*, *Desired Rent Income Fraction*, and *Min Investor Percentile*. All values in the *GP-emulated Experiment* section are averages over 10 independent runs of the emulated Metropolis-Hastings algorithm with standard deviations in brackets. The same ensemble of runs is also used to calculate the summary statistics in the *Performance Metrics* section. All execution time measurements are expressed as a percentage of the approximate time required to generate an equivalent number of posterior samples without emulation.

Now, referring to Table 5.1, which outlines the results of the above investigation[27], we see that there are a number of aspects worth highlighting. Firstly, regarding the quality of the posterior approximation, we observe that although the TV distances are, on average, higher than those obtained for the previously considered simple models, they remain below our threshold of 0.15, indicating that the procedure has succeeded in emulating the original experiment[28]. Further, a more detailed analysis of the obtained posterior means and standard deviations also supports this conclusion by demonstrating strong agreement between the actual and emulated posterior point estimates. It is therefore natural to conclude that despite the increased difficulty of this example, we are still able to obtain reasonable

[27] We limit ourselves to $\epsilon = 0.35$ as a result of the increased complexity of this example, but it is reasonable to expect that the quality of the approximation would improve for stricter thresholds. Additionally, as a result of possible overfitting when employing the Matern kernel, we have also opted to focus exclusively on the RBF kernel.

[28] Only a single run of the procedure resulted in a TV distance greater than 0.15.

posterior approximations[29].

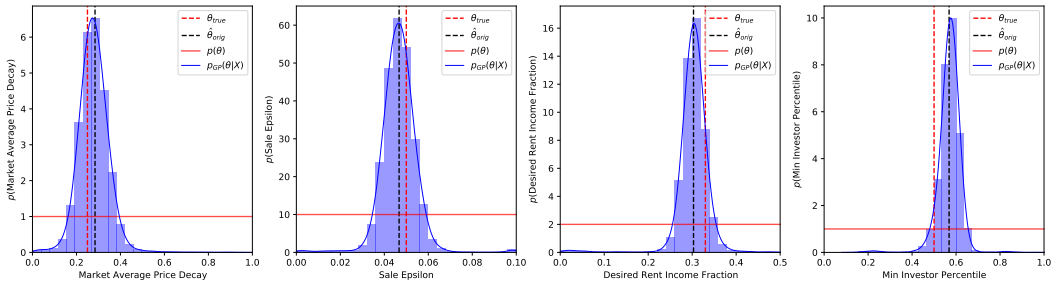


Figure 5.5: Marginal posterior distributions, $p_{GP}(\theta|\boldsymbol{X})$, for the housing market model obtained by pooling together samples from 10 independent runs of the GP-emulated Metropolis-Hastings algorithm with an RBF kernel and $\epsilon = 0.35$. $\hat{\theta}_{orig}$, the posterior mean obtained in the original, non-emulated experiments, is plotted for comparison.

Similar findings also apply when assessing the computational savings. While a greater number of likelihood evaluations[30] is required when compared to the Brock and Hommes (1998) model example, which also involved 4 free parameters, we still employ only a fraction of the original 25000 evaluations. Additionally, the overall computational time required is only about 10% of the original requirement, suggesting that the procedure has again succeeded in substantially reducing the computational burden of the estimation procedure. It is especially important to note that, in the context of the housing market model, this tenfold improvement corresponds to a reduction in computation time from several days to only a few hours in many runs.

Interestingly, it turns out that most of the computational time spent during the running of the emulated Metropolis-Hastings algorithm is devoted to solving the GP hyperparameter and related optimisation problems, which become more expensive as the number of training examples is increased. It is worth pointing out, however, that our settings for these procedures are extremely conservative. As stated in Appendix 5.B.3, we optimise the GP hyperparameters after the addition of every new training point and re-run the optimisation procedure a total of 10 times. It would not be unreasonable to expect that updating the hyperparameters every 5-10 iterations instead (after a reasonable number of evaluations, e.g. 200 or 250) and considering a smaller number of restarts would result in minimal sacrifices in accuracy, but even greater increases in computational efficiency[31]. Along these lines, it would therefore be interesting to, in future work, consider streamlining the training process as outlined above.

5.4 Practical Considerations

When presenting our results, we briefly alluded to the fact that, for certain estimation problems, the approach of Järvenpää and Corander (2021) might not be suitable and we

[29] See Figure 5.5 for an illustration of the marginals obtained by pooling all of the samples obtained over the 10 runs of the algorithm.

[30] When performing a given likelihood evaluation, we generate the $R = 50$ required model Monte Carlo replications in parallel on a small-scale, 64-core cluster.

[31] Järvenpää and Corander (2021) make similar suggestions.

would therefore, unfortunately, need to revert to the non-emulated sampling procedure. We now present a more complete examination of these limitations.

As previously discussed, successful applications of the GP-emulated sampling procedure usually follow a pattern in which the first few hundred iterations are relatively time-consuming as a result of the GP being iteratively trained and the posterior region still needing to be approached. After this initial set of iterations, however, a type of convergence is observed in which almost all subsequent iterations of the emulated sampling algorithm can be performed without any need to retrain the GP, leading to a significant overall speed-up. We illustrate this graphically in Figure 5.6 for the random walk model, where we observe that, during the first 1000 iterations, the amount of computational time required is relatively large, before falling substantially for most iterations thereafter.

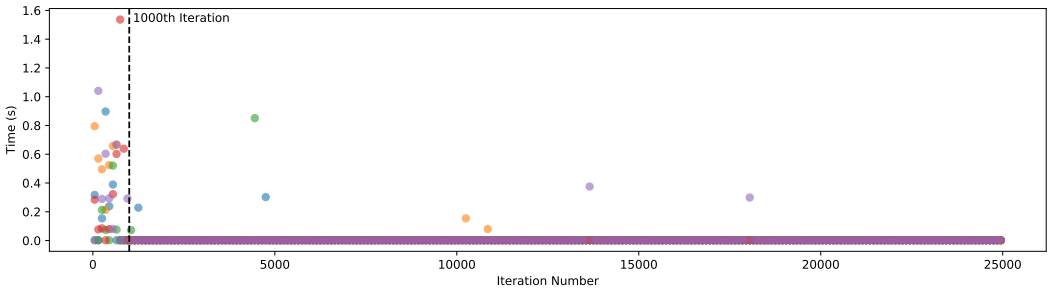


Figure 5.6: Iteration time measurements taken over 5 separate runs of the adaptive Metropolis-Hastings algorithm (random walk experiment, RBF kernel, and $\epsilon = 0.25$). Each point represents the average computational time over 100 iterations and different colours indicate different runs.

In cases for which the GP is unable to provide a reasonable approximation to the posterior, this convergent behaviour is not observed. Rather, additional training examples are continuously added to the GP training set for almost every iteration of the adaptive Metropolis-Hastings algorithm in order to keep the decision error below ϵ. Järvenpää and Corander (2021) indicate that increased difficulty is primarily observed in the case of posteriors with long tails[32] and we find that, despite the modifications made to the algorithm, our implementation behaves similarly. Intuitively, this occurs due to the fact that it is often very difficult to comprehensively explore the posterior in such cases. As a result, the GP is unable to learn a sufficiently accurate model for the posterior and thus produces high decision errors for almost all iterations of the algorithm.

While this is certainly a relevant limitation of the approach, we believe that the emulation methodology introduced by Järvenpää and Corander (2021) is still of great practical value in general. Firstly, it should be noted that although problematic posteriors will, of course, be encountered from time to time, posteriors do, in many cases, tend towards a Gaussian distribution as a result of the Bayesian central limit theorem. Therefore, even when considering a complex case like the INET Oxford housing market model, we still find the vast majority of the model's parameters to be associated with well-behaved marginal

[32]Similar behaviours might also be observed for highly-skewed posteriors (Järvenpää and Corander 2021). Note, however, that we found mild skewness to not be of much concern (see, for example, the *Min Investor Percentile* parameter of the housing market model in Figure 5.5).

posteriors and believe this to likely be the case for a large number of possible problems of interest.

Additionally, it should also be noted that constructing an accurate surrogate for long-tailed posteriors is a difficult problem in general; many of the global surrogate approaches introduced in Section 5.1.1 would also struggle to construct an effective metamodel in such cases. Unlike these techniques, however, the methodology introduced by Järvenpää and Corander (2021) is more robust to so-called 'silent failure'. Indeed, most global surrogate approaches would, even when the resulting approximation is poor, return a misleading final result. This is extremely problematic in most practical contexts as one would usually not know a-priori that the posterior may be challenging to model. Since non-convergence of the algorithm introduced by Järvenpää and Corander (2021) is straightforward to detect automatically[33], this becomes less of a concern. Further, even if convergence is not observed, previous iterations of the sampler may still be retained and the process continued without emulation, since we can be somewhat confident that, even if the use of a surrogate has failed to yield computational savings, the previously constructed sample sets are at least still valid as a result of the decision error having been kept below the threshold.

Conclusion

Throughout this chapter, we have demonstrated how the recently-introduced surrogate modelling approach of Järvenpää and Corander (2021) may be employed to greatly reduce the number of likelihood evaluations required when applying the Bayesian inference procedure we proposed in Chapters 3 and 4. Overall, we find that, in a number of different contexts, the use of emulation yields significant computational savings, requiring, in the worst case, a mere 10% of the running time associated with the non-emulated experiments, even when considering a complex ABM of the UK housing market. Additionally, even in cases for which emulation is not suitable, failure of the GP-emulated Metropolis-Hastings algorithm is usually straightforward to diagnose and one is therefore less likely to, as previously discussed, obtain flawed insights as a result of the 'silent failures' often encountered by competing surrogate modelling methodologies.

Although one might be tempted to view our exercises as a straightforward application of this existing methodology, there are a number of aspects worth highlighting. Firstly, Järvenpää and Corander (2021) consider a simple, summary statistic-based likelihood approximation technique, while we, on the other hand, consider a more general, summary statistic-free alternative. This is particularly important, as the GP-emulated Metropolis-Hastings framework, as it was originally introduced, leverages the fact that this simple likelihood construction approach allows for the consideration of a heterogenous GP observation noise intensity, σ_y^i, that can be easily determined through a bootstrapping procedure. In our case (and indeed for many other sophisticated likelihood approximation methods), however, this is intractable and one is instead forced to consider a homogenous noise intensity that is set during the hyperparameter optimisation procedure. Despite this modification, we find that the emulated Metropolis-Hastings algorithm still performs well and demonstrates behaviours consistent with those found in the original investigation. It therefore appears that the assumption of heterogeneous observation noise is not a critical component of the

[33] As an example, one could define a threshold, e.g. the 1000th iteration, and abort the emulation process if more than 75% of the next 1000 iterations (up to the 2000th), require GP retraining. Alternatively, one could also define a maximum evaluation budget and abandon the procedure if this is exceeded.

methodology and that it can therefore be applied to a wider class of likelihood approximators than perhaps originally seemed plausible.

Finally, it should also be kept in mind that, by considering the multi-sample Metropolis-Hastings algorithm introduced by Griffin and Walker (2013) rather than a standard, single-sample variant, we also observe what appears to be an increased robustness to the starting point of the MCMC sampler.

5.A Total Variation Distance

The TV distance is a simple metric that quantifies the degree of dissimilarity between two probability distributions. Formally, it is given by

$$TV(\pi, \pi') = \frac{1}{2} \int_{\boldsymbol{\Theta}} |\pi(\boldsymbol{\theta}) - \pi'(\boldsymbol{\theta})| d\boldsymbol{\theta}, \tag{5.24}$$

where π and π' are PDFs defined on $\boldsymbol{\Theta}$. A key advantage over other similar metrics is the fact that it takes values in the interval $[0, 1]$, with $TV = 0 \iff \pi = \pi'$, implying that interpretation is relatively straightforward. In particular, Järvenpää and Corander (2021) indicate that TV distances in the range of approximately 0.1 can be taken as evidence of high-quality emulated posteriors and they additionally describe examples with TV distances of up to around 0.15 as corresponding to reasonable approximations. Along these lines, when interpreting the TV distances obtained in our experiments, we consider emulated posteriors associated with TV distances less than 0.15 as being reasonable[34].

Of course, such a metric would only be useful if it could be easily estimated. Fortunately, the calculation of a reasonable estimator is also straightforward in our context. Assuming that one has obtained samples (e.g. from the output of an MCMC algorithm), π (π') can, in the univariate case, be approximated by dividing its support, Θ, into N_{bins} equally-sized bins[35], counting the number of samples from π (π') that fit into each bin, and normalising these counts by the total number of samples. The approximate TV distance would then be given by

$$\hat{TV}(\pi, \pi') = \frac{1}{2} \sum_{i=1}^{N_{bins}} |\pi_i - \pi'_i|, \tag{5.25}$$

where π_i is the normalised count for bin i and π'_i is analogously defined. In the multivariate case, we would simply repeat the above procedure for each dimension and take the average, leading to an overall TV distance estimate based on a set of marginal TV distances[36].

5.B Additional Experimental Details

To conclude this chapter's appendix, we provide a brief summary of additional details related to the surrogate construction process that were not covered in the main text.

5.B.1 Dealing with Invalid Likelihood Evaluations

In Section 5.1.1, we briefly alluded to the fact that it may not be possible to evaluate the simulation-based likelihood for certain parameter combinations as a result of model crashes. When applying GP-emulated MCMC, this may occur during either of two stages. Firstly, it may occur during the initial training phase before the main Metropolis-Hastings run. In this case, we simply remove the invalid samples from $\mathcal{D}_0$ and train the GP surrogate on the remaining training examples. While this would be extremely problematic for a global

[34] Additionally, while not used as an explicit threshold, we also consider the proximity of the obtained TV distances to 0.1 when making our assessment.

[35] We consider $N_{bins} = 100$ in all cases.

[36] In principle, the TV distance could also be calculated using the joint PDFs. These would, however, be considerably more difficult to estimate, leading us to consider the marginal-average approach suggested by Järvenpää and Corander (2021) instead.

surrogate, the approach of Järvenpää and Corander (2021) is not affected by the removal of samples in these regions. This is because areas of the parameter space that were not well-covered by the training set would be associated with high $E_{\hat{\gamma}}$ values and the GP-emulated Metropolis-Hastings algorithm would therefore not accept any candidate parameter samples until additional likelihood evaluation locations had been generated and the error had been brought below the required tolerance.

Of course, it is also possible that when trying to reduce the decision error, $\boldsymbol{\theta}_{opt}$ itself may also produce an invalid likelihood, leading to a second case to be considered. In such situations, it is not possible to improve the surrogate through the consideration of additional likelihood evaluations and we must therefore revert to the original simulation-based likelihood and compute $P(\boldsymbol{X}|\boldsymbol{\theta})$ and $P(\boldsymbol{X}|\boldsymbol{\theta}')$ directly. Fortunately, even if these direct evaluations were to fail, they can be dealt with relatively easily using the standard approach of assigning arbitrarily small log-likelihoods and computing the acceptance probability as per usual.

5.B.2 Rescaling of the Training Data

As in the case of many other machine learning methods, GPs can, in some cases, be sensitive to differences in the scale of features and outputs included in the training data. For this reason, GPs often deliver superior performance when each variable in the training data is rescaled to have zero mean and unit variance (Murphy 2012). We thus apply a similar normalisation scheme to that employed in Section 3.A.3 and rescale each training set entry, $(\boldsymbol{\theta}_i, y_i)$, according to

$$\begin{aligned} \tilde{\boldsymbol{\theta}}_i &= diag(\hat{\boldsymbol{\sigma}}_\theta)^{-1}(\boldsymbol{\theta}_i - \hat{\boldsymbol{\mu}}_\theta), \\ \tilde{y}_i &= \hat{\sigma}_y^{-1}(y_i - \hat{\mu}_y), \end{aligned} \tag{5.26}$$

where $\hat{\boldsymbol{\mu}}_\theta$ and $\hat{\boldsymbol{\sigma}}_\theta$ are vectors that contain estimates of the mean and standard deviation respectively for each dimension of $\boldsymbol{\theta}$ across the training set and $\hat{\mu}_y$ and $\hat{\sigma}_y$ are analogously defined.

When performing inference using the trained GP, the candidate set of parameter combinations at which the posterior is to be evaluated, $\boldsymbol{\theta}^*$, should be rescaled as in Eqn. 5.26, using the same $\hat{\boldsymbol{\mu}}_\theta$ and $\hat{\boldsymbol{\sigma}}_\theta$ values originally estimated from the training data. The obtained posterior mean vector, $\tilde{\boldsymbol{\mu}}^*$, and covariance matrix, $\tilde{\boldsymbol{\Sigma}}^*$, should then be rescaled such that they reflect the original range of variation of f, yielding

$$\begin{aligned} \boldsymbol{\mu}^* &= \hat{\sigma}_y \tilde{\boldsymbol{\mu}}^* + \hat{\mu}_y, \\ \boldsymbol{\Sigma}^* &= \hat{\sigma}_y^2 \tilde{\boldsymbol{\Sigma}}^*. \end{aligned} \tag{5.27}$$

5.B.3 Optimisation

Optimisation methods are required during two key steps of the surrogate construction procedure: the determination of appropriate GP hyperparameters and the selection of which new parameter combinations, $\boldsymbol{\theta}_{opt}$, to add to the training set during attempts to reduce the Metropolis-Hastings decision error. In both cases, we use a limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm[37], repeat the optimisation procedure $N_{restarts}$ times

[37]This is implemented in Python's SciPy library as a part of the *minimize* method. See `https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html` for additional details.

from different random initial guesses, and select the best (as measured by the objective function) optimiser obtained over the optimisation runs.

When maximising the log marginal likelihood, we constrain the hyperparameters to take values in the interval $[10^{-8}, \infty)$ and similarly consider random initial guesses for each dimension in the interval $[10^{-8}, 20]$. The only exception is σ_y, which we constrain[38] to take values between 0 and 1. Unless otherwise stated, $N_{restarts} = 10$ for all hyperparameter optimisation attempts.

When determining $\boldsymbol{\theta}_{opt}$, we derive both our constraints and the range of the initial guess from $\boldsymbol{\Theta}_{init}$ and set $N_{restarts} = 5$.

[38]This is also the range considered during the generation of the initial guess.

Conclusion

Throughout the preceding chapters of this thesis, we engage extensively with the problem of statistical inference in the context of economic and financial ABMs, both by providing a detailed examination of prior literature and by developing and applying our own novel approach.

We begin our analysis in Chapter 1, which provides a comprehensive literature review and identifies a number of key issues plaguing existing contributions. In particular, we highlight three problematic conventions in the field, including the absence of direct comparisons between existing methods, the lack of standard benchmarks against which to test new methodologies, and very few applications to more complex or computationally-expensive models. This ultimately leaves the community with a large set of potential methods to choose from, very little guidance on which methods to employ and in which circumstances, and, in many cases, significant difficulties when attempting to apply existing methods to large-scale models.

In response, Chapter 2, which acts as the first major contribution of this thesis, provides an extensive comparative exercise in which several prominent ABM calibration techniques are pitted against one another in a series of parameter recovery challenges. Interestingly, these exercises reveal that popular SMD methods, which construct and minimise sophisticated frequentist objective functions, often underperform when compared to the relatively simple, likelihood-based Bayesian approach introduced by Grazzini et al. (2017). Further, our results also demonstrate that the approach of Grazzini et al. (2017) may be applied to a large-scale model of the UK housing market and still yield reasonable results, ultimately leading us to conclude that further development of the aforementioned approach and Bayesian techniques in general (which were less popular at the time) is warranted[39].

This naturally leads to Chapter 3, where we examine the methodology introduced by Grazzini et al. (2017) more critically. In particular, we indicate that, despite its past successes, the approach makes very strong independence assumptions that are unlikely to hold in practice. To address these shortcomings, we introduce our own neural network-based likelihood approximation technique that relaxes these problematic assumptions and demonstrate how the resultant Bayesian estimation framework consistently outperforms the approach of Grazzini et al. (2017) when benchmarked through a series of experiments involving a variety of simple time series models. Finally, we also show that our proposed approach, much like that of Grazzini et al. (2017), may be applied to more complex examples, such as the housing market model considered in Chapter 2, and illustrate how significant improvements in estimation performance are also encountered in this more challenging context.

Following on from the above contribution, Chapter 4 builds upon the foundation estab-

[39] As discussed in Chapter 1, emphasis on Bayesian approaches has increased dramatically over the last year.

lished in Chapter 3 and demonstrates how our neural network-based likelihood approximation technique may be combined with the DIC and PSIS-LOO, two contemporary Bayesian approaches to predictive model selection, to facilitate the process of comparing ABMs to one another and to competing modelling frameworks. We apply these techniques extensively in the context of equity return data, both at inter- and intraday time scales. In the interday case, which has previously been investigated by other authors and involves the consideration of relatively simple models, we recover known results and reinforce existing findings regarding the competitiveness of ABMs relative to a benchmark ARMA-GARCH model. More importantly, however, we show that similar findings also apply at intraday time scales, which had previously not been as extensively investigated as a result of the complexity and computational expense associated with order book models. In particular, we provide a concrete example of the ability of the behavioural LOB model introduced by Chiarella and Iori (2002) to more accurately replicate the dynamics of intraday, high-frequency mid-price returns when benchmarked against a standard statistical model and demonstrate that the competitiveness of ABMs appears to largely stem from a general ability to more accurately model extreme price movements.

Finally, the thesis concludes with Chapter 5, which showcases how recent advances in surrogate modelling may be incorporated into the framework introduced in Chapters 3 and 4 to drastically reduce the associated computational costs. To elaborate, we show how the GP-emulated Metropolis-Hastings algorithm introduced by Järvenpää and Corander (2021) may be adapted and combined with our own neural network-based likelihood approximation technique to produce results of a similar quality to those presented in Chapter 3, while, at the same time, requiring less than 10% of the original computational time and only several hundred likelihood evaluations.

Overall, we believe that the preceding chapters, when viewed collectively, make an important contribution and provide a comprehensive approach to Bayesian inference for economic and financial ABMs that avoids the problem of needing to transform the considered data into sets of summary statistics, while, at the same time, remaining tractable when applied to more complex examples. Despite this success, however, it must be noted that we are by no means claiming that our method is definitive. Rather, we very much expect that some variation of the No Free Lunch Theorem applies and that it is almost inevitable that our approach will, in some cases, not be appropriate. A comprehensive approach to inference would therefore inevitably call for a variety of methods. Nevertheless, and as we have demonstrated throughout this thesis, we believe that our approach would certainly form a useful part of the general ABM toolbox and provide useful insights in a wide variety of cases.

Bibliography

S. Alfarano, T. Lux, and F. Wagner. Estimation of agent-based models: The case of an asymmetric herding model. *Computational Economics*, 26(1):19–49, 2005.

S. Alfarano, T. Lux, and F. Wagner. Estimation of a simple agent-based model of financial markets: An application to australian stock and foreign exchange data. *Physica A: Statistical Mechanics and its Applications*, 370(1):38–42, 2006.

S. Alfarano, T. Lux, and F. Wagner. Empirical validation of stochastic models of interacting agents. *The European Physical Journal B: Condensed Matter and Complex Systems*, 55 (2):183–187, 2007.

S. Alfarano, T. Lux, and F. Wagner. Time variation of higher moments in a financial market with heterogeneous agents: An analytical approach. *Journal of Economic Dynamics and Control*, 32(1):101–136, 2008.

G. Attanasi, S. Centorrino, and I. Moscati. Over-the-counter markets vs. double auctions: A comparative experimental study. *Journal of Behavioral and Experimental Economics*, 63:22–35, 2016.

R. Axtell. Zipf distribution of u.s. firm sizes. *Science*, 293:1818–1820, 2001.

R. Baptista, J. Farmer, M. Hinterschweiger, K. Low, D. Tang, and A. Uluc. Macroprudential policy in an agent-based model of the uk housing market. *Bank of England Staff Working Paper*, 619, 2016.

S. Barde. Direct comparison of agent-based models of herding in financial markets. *Journal of Economic Dynamics and Control*, 73:326–353, 2016.

S. Barde. A practical, accurate, information criterion for nth order markov processes. *Computational Economics*, 50:281–324, 2017.

S. Barde and S. van der Hoog. An empirical validation protocol for large-scale agent-based models. *Studies in Economics, School of Economics, University of Kent*, 2017.

L. Bauwens and P. Giot. *Econometric Modelling of Stock Market Intraday Activity*. Springer, 2001.

A. Berg, R. Meyer, and J. Yu. Deviance information criterion for comparing stochastic volatility models. *Journal of Business and Economic Statistics*, 22(1):107–120, 2004.

N. Bertschinger and I. Mozzhorin. Bayesian estimation and likelihood-based comparison of agent-based volatility models. *Journal of Economic Interaction and Coordination*, https://doi.org/10.1007/s11403-020-00289-z, 2020.

C. Bianchi, P. Cirillo, M. Gallegati, and P. Vagliasindi. Validating and calibrating agent-based models: A case study. *Computational Economics*, 30(3):245–264, 2007.

C. Bishop. Mixture density networks. Technical report, Aston University, 1994.

BMLL. Bmll data lab, 2021. URL `https://bmlltech.com/our-range-of-products/bmll-data-lab/`.

W. Brock and C. Hommes. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8-9):1235–1274, 1998.

P. C. Bürkner, J. Gabry, and A. Vehtari. Approximate leave-future-out cross-validation for bayesian time series models. *Journal of Statistical Computation and Simulation*, 10.1080/00949655.2020.1783262, 2020.

F. Campolongo, J. Cariboni, and A. Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental Modelling and Software*, 22(10):1509–1518, 2007.

E. Carrella. No free lunch when estimating simulation parameters. *Journal of Artificial Societies and Social Simulation*, 24(2):7, 2021.

G. Celeux, F. Forbes, C. Robert, and D. Titterington. Deviance information criteria for missing data models. *Bayesian Analysis*, 1(4):651–674, 2006.

J. Chan and A. Grant. On the observed-data deviance information criterion for volatility modeling. *Journal of Financial Econometrics*, 14(4):772–802, 2016.

S. Chen. Agent-based computational macroeconomics: A survey. In T. Terano, H. Deguchi, and K. Takadama, editors, *Meeting the Challenge of Social Problems via Agent-Based Simulation*, pages 141–170. Springer-Verlag, 2003.

Z. Chen and T. Lux. Estimation of sentiment effects in financial markets: A simulated method of moments approach. *Computational Economics*, 52(3):711–744, 2018.

C. Chiarella and G. Iori. A simulation analysis of the microstructure of double auction markets. *Quantitative Finance*, 2(5):346–353, 2002.

S. Cincotti, M. Raberto, and A. Teglio. Credit money and macroeconomic instability in the agent-based model and simulator eurace. *Economics: The Open-Access, Open-Assessment E-Journal*, 4:1–32, 2010.

C. Collver. An application of agent-based modeling to market structure policy: the case of the u.s. tick size pilot program and market maker profitability. Technical report, U.S. Securities and Exchange Commission, 2017.

R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223–236, 2001.

R. Cont, S. Stoikov, and R. Talreja. A stochastic model for order book dynamics. *Operations Research*, 58(3):549–563, 2010.

L. Dall'Amico, A. Fosset, J. Bouchaud, and M. Benzaquen. How does latent liquidity get revealed in the limit order book? *Journal of Statistical Mechanics: Theory and Experiment*, 013404, 2019.

G. Debreu. Excess-demand functions. *Journal of Mathematical Economics*, 1:15–21, 1974.

D. Delli Gatti and J. Grazzini. Rising to the challenge: Bayesian estimation and forecasting techniques for macroeconomic agent based models. *Journal of Economic Behavior and Organization*, 178:875–902, 2020.

J. Donier and J. P. Bouchaud. From walras' auctioneer to continuous time double auctions: a general dynamic theory of supply and demand. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(12):123406, 2016.

G. Dosi, G. Fagiolo, and A. Roventini. Schumpeter meeting keynes: A policy-friendly model of endogenous growth and business cycles. *Journal of Economic Dynamics and Control*, 34(9):1748–1767, 2010.

G. Dosi, M. Pereira, and M. Virgillito. On the robustness of the fat-tailed distribution of firm growth rates: a global sensitivity analysis. *Journal of Economic Interaction and Coordination*, 2017.

D. Easley, M. Lopez de Prado, and M. O'Hara. The volume clock: Insights into the high-frequency paradigm. *Journal of Portfolio Management*, 39(1):19–29, 2012.

R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference On Learning Theory*, 2016.

A. Fabretti. On the problem of calibrating an agent based model for financial markets. *Journal of Economic Interaction and Coordination*, 8(2):277–293, 2013.

G. Fagiolo and A. Roventini. Macroeconomic policy in dsge and agent-based models redux: New developments and challenges ahead. *Journal of Artificial Societies and Social Simulation*, 20(1):1, 2017.

G. Fagiolo, M. Guerini, F. Lamperti, A. Moneta, and A. Roventini. Validation of agent-based models in economics and finance. In C. Beisbart and N. Saam, editors, *Computer Simulation Validation. Simulation Foundations, Methods and Applications*. Springer, 2019.

J. Farmer and D. Foley. The economy needs agent-based modelling. *Nature*, 460:685–686, 2009.

J. Farmer and S. Joshi. The price dynamics of common trading strategies. *Journal of Economic Behavior and Organization*, 49(2):149–171, 2002.

R. 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.

R. Franke and F. Westerhoff. Structural stochastic volatility in asset pricing dynamics: Estimation and model contest. *Journal of Economic Dynamics and Control*, 36(8):1193–1211, 2012.

A. Gaier and D. Ha. Weight agnostic neural networks. In *Advances in Neural Information Processing Systems*, 2019.

J. Geanakoplos and J. Farmer. The virtues and vices of equilibrium and the future of financial economics. *Complexity*, 14(3):11–38, 2008.

J. Geanakoplos, R. Axtell, J. Farmer, P. Howitt, B. Conlee, J. Goldstein, M. Hendrey, N. Palmer, and C. Yang. Getting at systemic risk via an agent-based model of the housing market. *American Economic Review*, 102(3):53–58, 2012.

A. E. Gelfand, D. K. Dey, and H. Chang. Model determination using predictive distributions with implementation via sampling-based methods. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics*. Oxford University Press, 1992.

A. Gelman and D. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511, 1992.

A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis*. Chapman and Hall, 2nd edition, 2004.

A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014.

M. Gilli and P. Winker. A global optimization heuristic for estimating agent based models. *Computational Statistics and Data Analysis*, 42(3):299–312, 2003.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

M. D. Gould, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison. Limit order books. *Quantitative Finance*, 13(11):1709–1742, 2013.

C. Gourieroux, A. Monfort, and E. Renault. Indirect inference. *Journal of Applied Econometrics*, 8:S85–S118, 1993.

R. Gramacy. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Chapman & Hall, 2020.

J. Grazzini. Analysis of the emergent properties: Stationarity and ergodicity. *Journal of Artificial Societies and Social Simulation*, 15(2):7, 2012.

J. Grazzini and M. Richiardi. Estimation of ergodic agent-based models by simulated minimum distance. *Journal of Economic Dynamics and Control*, 51:148–165, 2015.

J. Grazzini, M. Richiardi, and M. Tsionas. Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control*, 77:26–47, 2017.

J. Griffin and S. Walker. On adaptive metropolis–hastings methods. *Statistics and Computing*, 23(1):123–134, 2013.

M. Guerini and A. Moneta. A method for agent-based models validation. *Journal of Economic Dynamics and Control*, 82:125–141, 2017.

L. Hansen and J. Heckman. The empirical foundations of calibration. *Journal of Economic Perspectives*, 10(1):87–104, 1996.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

S. Jacob Leal, M. Napoletano, A. Roventini, and G. Fagiolo. Rock around the clock: An agent-based model of low- and high-frequency trading. *Journal of Evolutionary Economics*, 26(1):49–76, 2016.

M. Järvenpää and J. Corander. Approximate bayesian inference from noisy likelihoods with gaussian process emulated mcmc. *arXiv*, 2104.03942, 2021.

A. Kaveh. Particle swarm optimization. In *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, chapter 2, pages 11–43. Springer-Verlag, 2017.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, May 7-9*, 2015.

P. Knysh and Y. Korkolis. blackbox: A procedure for parallel optimization of expensive black-box functions. *arXiv*, 1605.00998, 2016.

A. Kononovicius and J. Ruseckas. Order book model with herd behavior exhibiting long-range memory. *Physica A: Statistical Mechanics and its Applications*, 525:171–191, 2019.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

J. Kukacka and J. Barunik. Estimation of financial agent-based models with simulated maximum likelihood. *Journal of Economic Dynamics and Control*, 85:21–45, 2017.

D. Ladley. Zero intelligence in economics and finance. *The Knowledge Engineering Review*, 27(2):273–286, 2012.

F. Lamperti. An information theoretic criterion for empirical validation of simulation models. *Econometrics and Statistics*, 5:83–106, 2018a.

F. Lamperti. Empirical validation of simulated models through the gsl-div: an illustrative application. *Journal of Economic Interaction and Coordination*, 13(1):143–171, 2018b.

F. Lamperti, A. Roventini, and A. Sani. Agent-based model calibration using machine learning surrogates. *Journal of Economic Dynamics and Control*, 90:366–389, 2018.

B. LeBaron. Agent-based computational finance. In L. Tesfatsion and K. Judd, editors, *Handbook of Computational Economics*, volume 2, chapter 24, pages 1187–1233. Elsevier, 2006.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Z. Lu, M. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, volume 31, 2017.

T. Lux. Estimation of agent-based models using sequential monte carlo methods. *Journal of Economic Dynamics and Control*, 91:391–408, 2018.

T. Lux. Bayesian estimation of agent-based models via adaptive particle markov chain monte carlo. *Computational Economics*, 10.1007/s10614-021-10155-0, 2021.

T. Lux and R. Zwinkels. Empirical validation of agent-based models. *SSRN*, 2926442, 2017.

C. Macal and M. North. Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4(3):151–162, 2010.

A. Mandes. Impact of electronic liquidity providers within a high-frequency agent-based modeling framework. *Computational Economics*, 55:407–450, 2020.

R. Mantel. On the characterization of aggregate excess-demand. *Journal of Economic Theory*, 7:348–353, 1974.

I. Mastromatteo, B. Tóth, and J. Bouchaud. Agent-based models for latent liquidity and concave price impact. *Physical Review E*, 89:042805, 2014.

D. McFadden. A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, 57(5):995–1026, 1989.

D. Mishkin and J. Matas. All you need is a good init. In *International Conference on Learning Representations*, 2016.

M. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

K. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

R. M. Navarro and H. Larralde. A detailed heterogeneous agent model for a single asset financial market with trading via an order book. *PLoS One*, 0170766, 2017.

A. Ntakaris, M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37:852–866, 2018.

K. Oh and K. Jung. Gpu implementation of neural networks. *Pattern Recognition*, 37(6): 1311–1314, 2004.

E. Panayi, M. Harman, and W. Wetherilt. Agent-based modelling of stock markets using existing order book data. In F. Giardini and F. Amblard, editors, *Multi-Agent-Based Simulation XIII*, pages 101–114. Springer-Verlag, 2013.

G. Papamakarios and I. Murray. Fast ϵ-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

D. Platt. A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, 113:103859, 2020.

D. Platt. Bayesian estimation of economic simulation models using neural networks. *Computational Economics*, 10.1007/s10614-021-10095-9, 2021.

D. Platt and T. Gebbie. Can agent-based models probe market microstructure? *Physica A: Statistical Mechanics and its Applications*, 503:1092–1106, 2018.

L. Prechelt. Early stopping - but when? In G. Orr and K. Müller, editors, *Neural Networks: Tricks of the Trade*. Springer, 1998.

T. Preis, S. Golke, W. Paul, and J. J. Schneider. Multi-agent-based order book model of financial markets. *Europhysics letters*, 75(3):510–516, 2006.

T. Preis, S. Golke, W. Paul, and J. J. Schneider. Statistical analysis of financial returns for a multiagent order book model of asset trading. *Physical Review E*, 76:016108, 2007.

L. Price, C. Drovandi, and D. Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

C. Rasmussen and K. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

M. Recchioni, G. Tedeschi, and M. Gallegati. A calibration procedure for analyzing stock price dynamics in an agent-based framework. *Journal of Economic Dynamics and Control*, 60:1–25, 2015.

R. Regis and C. Shoemaker. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31(1):153–171, 2005.

C. Robert and G. Casella. *Introducing Monte Carlo Methods with R*. Springer-Verlag, 2010.

J. Rothfuss, F. Ferreira, S. Walther, and M. Ulrich. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv*, 1903.00954, 2019.

I. Safran and O. Shamir. Depth-width tradeoffs in approximating natural functions with neural networks. In *International Conference on Machine Learning*, volume 34, 2017.

I. Salle and Yildizoglu. Efficient sampling and meta-modeling for computational economic models. *Computational Economics*, 44(4):507–536, 2014.

I. Schwartz. Estimation of agent-based models: Testing and applying a simulated joint moment approach. *Working Paper, University of Bamberg*, 2021.

D. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley and Sons, 2015.

S. Sheehan and Y. S. Song. Deep learning for population genetic inference. *PLoS Computational Biology*, 12(3):e1004845, 2016.

T. Shiono. Estimation of agent-based models using bayesian deep learning approach of bayesflow. *Journal of Economic Dynamics and Control*, 125:104082, 2021.

S. Sisson, Y. Fan, and M. Beaumont. Overview of approximate bayesian computation. In S. Sisson, Y. Fan, and M. Beaumont, editors, *Handbook of Approximate Bayesian Computation*, chapter 1. CRC Press, 2018.

E. Smith, J. D. Farmer, L. Gillemot, and L. Krishnamurthy. Statistical theory of the continuous double auction. *Quantitative Finance*, 3(6):481–514, 2003.

H. Sonnenschein. Do walras' identity and continuity characterize the class of community excess-demand functions? *Journal of Economic Theory*, 6:345–354, 1973.

D. Spiegelhalter, N. Best, B. Carlin, and A. van der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

D. Spiegelhalter, N. Best, B. Carlin, and A. van der Linde. The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):485–493, 2014.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012.

S. Tavare, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

L. Tesfatsion and K. L. Judd, editors. *Handbook of Computational Economics*, volume 2. Elsevier, 2006.

J. W. van de Meent, B. Paige, H. Yang, and F. Wood. An introduction to probabilistic programming. *arXiv*, 1809.10756, 2018.

A. Vehtari, A. Gelman, and J. Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 27(5):1413–1432, 2017.

A. Vehtari, P. C. Bürkner, and J. Gabry. Leave-one-out cross-validation for non-factorizable models. Available at `mc-stan.org/loo/articles/loo2-non-factorizable.html#references`, 2019.

P. Vytelingum, D. Cliff, and N. R. Jennings. Strategic bidding in continuous double auctions. *Artificial Intelligence*, 172(14):1700–1729, 2008.

L. Walras. *Elements of Pure Economics*. Routledge, 2013.

S. Watanabe. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11: 3571–3594, 2010.

F. Willems, Y. Shtarkov, and T. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, IT–41:653–664, 1995.

S. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466: 1102–1104, 2010.