Hybrid Monte Carlo Methods In Machine Learning:

Stochastic Volatility Methods, Shadow Hamiltonians, Adaptive Approaches and Variance Reduction Techniques

by

Wilson Tsakane Mongwe

A thesis submitted at the Faculty of Engineering and Built Environment in fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in the subject

Electrical and Electronic Engineering

at the

SUPERVISOR: Prof. Tshilidzi Marwala

CO-SUPERVISOR: Dr Rendani Mbuvha

February 2022

Declaration of Authorship

I declare that the thesis titled "*Hybrid Monte Carlo Methods In Machine Learning: Stochastic Volatility Methods, Shadow Hamiltonians, Adaptive Approaches and Variance Reduction Techniques*" is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I have not previously submitted this work, or part of it, for examination at the University of Johannesburg for another qualification or at any other higher education institution.

Name: Wilson Tsakane Mongwe

Student Number: 220085609

Signature: WT Mongwe

Date: 02 February 2022

Hybrid Monte Carlo Methods In Machine Learning:

Stochastic Volatility Methods, Shadow Hamiltonians, Adaptive Approaches and Variance Reduction Techniques

by

Wilson Tsakane Mongwe

Abstract

Markov Chain Monte Carlo (MCMC) methods are a vital inference tool for probabilistic machine learning models. A commonly utilised MCMC algorithm is the Hamiltonian Monte Carlo (HMC) method. HMC can efficiently explore the target posterior by taking into account first-order gradient information. This algorithm has been extended in various ways, including via the use of non-canonical Hamiltonian dynamics to create Magnetic Hamiltonian Monte Carlo (MHMC) and utilising integrator-dependent shadow Hamiltonians to create shadow HMC methods. MHMC utilises a magnetic field to more efficiently explore the target posterior compared to HMC. At the same time, shadow HMC methods can better scale to larger models without significant deterioration in the acceptance rates of the generated samples.

In this thesis, we present novel extensions to the MHMC algorithm by 1) using a random mass for the auxiliary momentum variable to mimic the behavior of quantum particles, 2) utilising partial momentum refreshment before generating the next sample, and 3) deriving the fourth-order modified Hamiltonian implied by the numerical integrator employed in MHMC to construct the novel shadow MHMC algorithm. The results reveal that these novel extensions to MHMC lead to enhanced sampling performance over MHMC across various targets and performance metrics.

We proceed to improve the sampling performance of Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) by using partial momentum refreshment before generating the next sample using the processed leapfrog integrator used in S2HMC. We further address the automatic tuning of the step size and trajectory length parameters of S2HMC by extending the No-U-Turn Sampler methodology to incorporate the processed leapfrog integrator. The automatic adaptation removes the need to manually tune these param-

eters and thus makes this method more accessible to non-expert users. The results show that this adaptive algorithm outperforms S2HMC on an effective sample size basis with minimal user intervention.

We further rely on results from the coupling theory of Hamiltonian chains to show that employing antithetic sampling in the MHMC and S2HMC algorithms produces lower variances and consequently higher effective sample sizes compared to the non-antithetic versions of these MCMC methods. The results show the overall benefits that can be derived from incorporating antithetic sampling in MCMC algorithms. Given that antithetic sampling makes minimal assumptions about the target posterior and is straightforward to implement, there seem to be minor hurdles to using it in practice.

The analysis in this thesis is performed on the Banana shaped distribution, the Merton jump-diffusion process calibrated to financial market data, multivariate Gaussian distributions of various dimensions, Neal's funnel density, Bayesian Logistic Regression, and Bayesian Neural Network benchmark problems as well as on South African municipal financial statement audit outcome data. To analyse the audit outcomes dataset, we employ a first-in-literature Bayesian inference approach that incorporates automatic relevance determination to identify important financial ratios in modeling audit outcomes. We find that *repairs and maintenance as a percentage of total assets ratio*, *current ratio*, *debt to total operating revenue*, *net operating surplus margin* and *capital cost to total operating expenditure ratio* are the important financial ratios when predicting local government audit outcomes. These results could be useful for various stakeholders to better understand the financial state of South African local government entities. Auditors could use them to improve the speed and overall quality of the audit process.

Keywords: Adaptive, Quantum-Inspired, Magnetic, Shadow, Hamiltonian Monte Carlo, Partial Momentum Refreshment, Antithetic Sampling, Machine Learning, Social Impact.

Supervisor : Prof. Tshilidzi Marwala
Co-supervisor : Dr. Rendani Mbuvha
School : Electrical and Electronic Engineering Science

"Education is the most powerful weapon which you can use to change the world"

– Nelson Mandela

Acknowledgements

I want to thank my supervisory team of Prof. Tshildzi Marwala and Dr Rendani Mbuvha. Without your advice and encouragement during this thesis, there would be no thesis to submit! My gratitude also goes out to Prof. Katherine Malan, who has been my research mentor throughout this thesis.

I want to acknowledge Google Research for providing me with financial support throughout this thesis through awarding me a Google PhD fellowship in Machine Learning. I also want to acknowledge the Center for High Performance Computing (CHPC) at the Council of Scientific and Industrial Research (CSIR) South Africa for providing me with the resources that I used to perform the computations in this thesis.

I want to thank my family, especially my wife Gavaza and my children Vuthlari and Vutivi, who have been very supportive and patient through the high and lows of producing this manuscript. I would also like to acknowledge my high school teachers, friends, and mentors who have offered consistent support throughout my university career.

Last but not least, I would like to thank the all mighty God who has ordered my steps to be where I am today. All the glory be to God! It is all by the grace of God.

In loving memory of Nyanisi Nkiyasi Mongwe.

Contents

Acronyms

A-MHMC Antithetic Magnetic Hamiltonian Monte Carlo.

A-S2HMC Antithetic Separable Shadow Hamiltonian Hybrid Monte Carlo.

AG-SA Auditor General of South Africa.

A-HMC Antithetic Hamiltonian Monte Carlo.

ARD Automatic Relevance Determination.

AUC Area Under The Receiver Operating Curve.

BCH Baker-Campbell-Hausdorff.

BLR Bayesian Logistic Regression.

BNN Bayesian Neural Network.

ESS Effective Sample Size.

FSF Financial Statement Fraud.

HMC Hamiltonian Monte Carlo.

JDP Jump-Diffusion Process.

JS2HMC Jittered S2HMC.

JSE Johannesburg Stock Exchange.

MALA Metropolis Adjusted Langevin Algorithm.

MCMC Markov Chain Monte Carlo.

MH Metropolis-Hastings.

MHMC Magnetic Hamiltonian Monte Carlo.

MLP Multilayer Perceptron.

MSE Mean Square Error.

NUTS No-U-Turn Sampler.

PHMC Hamiltonian Monte Carlo With Partial Momentum Refreshment.

PMHMC Magnetic Hamiltonian Monte Carlo With Partial Momentum Refreshment.

PS2HMC Separable Shadow Hamiltonian Hybrid Monte Carlo With Partial Momentum Refreshment.

QIHMC Quantum-Inspired Hamiltonian Monte Carlo.

QIMHMC Quantum-Inspired Magnetic Hamiltonian Monte Carlo.

RMHMC Riemannian Manifold Hamiltonian Monte Carlo.

ROC Receiver Operating Curve.

S2HMC Separable Shadow Hamiltonian Hybrid Monte Carlo.

SHMC Shadow Hamiltonian Monte Carlo.

SMHMC Shadow Magnetic Hamiltonian Monte Carlo.

SOM Self Organising Map.

List of Symbols

N	Total number of samples generated
L	Trajectory length parameter in Hamiltonian samplers
ϵ	Step size parameter in Hamiltonian samplers
$\mathbf{w}$	Model parameters
D	Dimension of the model parameters $\mathbf{w}$
$\mathbf{p}$	Auxiliary momentum variable that is independent of $\mathbf{w}$
$\mathrm{U}(\mathbf{w})$	Potential energy evaluated at $\mathbf{w}$
$\mathrm{K}(\mathbf{p})$	Kinetic energy evaluated at $\mathbf{p}$
$\mathrm{H}(\mathbf{w}, \mathbf{p})$	Hamiltonian evaluated at $\mathbf{w}$ and $\mathbf{p}$
$\mathbf{M}$	Mass matrix for the momentum auxiliary variable $\mathbf{p}$
$\mathcal{N}(a, b)$	Normal distribution with mean a and variance b
$\mathbf{P}_{\mathbf{M}}(\mathbf{M})$	Probability distribution of the mass matrix $\mathbf{M}$
$\mathbf{G}$	Magnetic field in **MHMC**
β	Volatility of volatility parameter in **QIHMC**
ρ	Momentum refreshment parameter
$\hat{R}$	Potential scale reduction factor.

List of Figures

List of Algorithms

List of Tables

Chapter 1

Introduction

1.1 Background

Machine learning technologies have been successfully deployed in various industries including self-driving cars, health, and weather forecasting among others [44, 12, 87, 143]. Deep learning is a class of machine learning models that has fostered the growth of artificial intelligence in industry and is a crucial enabler of the fourth industrial revolution [142]. These machine learning models are typically trained within a frequentist paradigm using gradient and evolutionary-based optimisation techniques, key of which is gradient descent and its extensions [19].

A disadvantage of the frequentist approach to training machine learning models is the inability of the trained models to naturally produce uncertainties in the parameters of the model and their resultant predictions [43, 50]. There has been recent progress in addressing this limitation, with the majority of the approaches typically being model specific [5, 88] or already having Bayesian equivalents [50]. Forming uncertainties around model predictions is particularly important for complex and mission-critical systems such as self-driving cars, medical diagnosis of patients, and a bank deciding whether or not to grant credit to a customer [1, 43]. In all these instances, various stakeholders would be interested in establishing the extent to which the model is certain about its forecasts [104].

The Bayesian framework offers a probabilistically robust approach to training ma-

chine learning models [1, 88, 50, 96]. This approach provides confidence levels in the model parameters and the resultant model predictions. It also allows one to perform comparisons between different models using the Bayesian evidence metric [96]. The Bayesian approach incorporates already existing knowledge before the data is observed through the prior distribution. The observations are then incorporated via the likelihood function, which generates the posterior distribution when combined with our prior beliefs. The posterior distribution is thus a refinement of our prior convictions in light of the newly observed data [140]. A fundamental limitation of training machine learning models within the Bayesian paradigm is that Bayesian approaches tend to be computationally expensive, and are thus difficult to scale to larger and more complex models [162, 52].

The Bayesian framework of inference is founded on Bayes theorem, which states that for a given model M with parameters $\mathbf{w}$ and data D, we have:

$$\underbrace{\mathbb{P}(\mathbf{w}|M)}_{\text{posterior}} = \frac{\overbrace{\mathbb{P}(D|\mathbf{w},M)}^{\text{likelihood}}\overbrace{\mathbb{P}(\mathbf{w}|M)}^{\text{prior}}}{\underbrace{\mathbb{P}(D|M)}_{\text{evidence}}} \tag{1.1}$$

where $\mathbb{P}(\mathbf{w}|M)$ is the posterior of the parameters given the model M, $\mathbb{P}(D|\mathbf{w},M)$ is the likelihood of the data D, $\mathbb{P}(\mathbf{w}|M)$ is the prior distribution for the parameters and $\mathbb{P}(D|M)$ is the Bayesian evidence for model M and can be utilised for model selection [104]. In this thesis, we aim to infer the posterior distribution of the parameters $\mathbb{P}(\mathbf{w}|M)$ for a given model so we can perform predictions using the model. The posterior distribution is typically not available in closed form, and thus, numerical techniques are often required to determine the posterior [96]. The common approaches for training models, that is, determining the posterior distribution, within the Bayesian framework are variational inference techniques and Markov Chain Monte Carlo (MCMC) algorithms.

Variational inference [70] and its extensions [72] are approximate inference techniques in that the posterior distribution is assumed to come from a particular family of simpler distributions, with the commonly used family being that of Gaussian distributions [52]. This allows the sampling problem to be converted into an optimisation problem in which the Kullback-Leibler divergence between the target posterior and the approximating

distribution is minimised [52]. This attribute makes variational methods typically much faster than MCMC methods [70, 52].

MCMC algorithms are preferable to variational techniques as they are guaranteed to converge to the correct target distribution if the sample size is sufficiently large [148, 149]. These methods are premised on building a Markov chain of samples that asymptotically, as the number of samples $n \to \infty$, converge to the desired equilibrium distribution. By definition, the samples generated by MCMC algorithms are auto-correlated, which means that they will have higher variance than classical Monte Carlo techniques. This branch of inference techniques was initially developed by physicists, with famous examples being the Metropolis-Hastings (MH) [65] algorithm of Metropolis and Hastings and Hamiltonian Monte Carlo (HMC) of Duane *et al.* [41]. These MCMC methods then later entered the field of computational statistics, where they are used today to sample from various complex probabilistic models [57]. This thesis focuses primarily on HMC and its variants with the aim of developing improved versions of HMC and its descendants.

The random walk MH algorithm forms the foundation of more complicated algorithms such as HMC and the Metropolis Adjusted Langevin Algorithm (MALA) [65, 124, 53]. The simplicity and robustness of this algorithm have resulted in it still being extensively used in various applications [168]. This method generates proposed samples from a Gaussian distribution whose mean is the last accepted state [65, 98]. This proposal exhibits random walk behaviour, which usually results in low sample acceptance rates and generates strongly correlated samples [124, 98]. In practice, the scale matrix of the Gaussian proposal distribution is typically treated as being diagonal, with all the dimensions having the same standard deviation $\sigma \geq 0$ [145]. Mongwe *et al.* [115] extend random walk MH by treating the scale matrix as being stochastic and allowing it to depend on the local geometry of the target. The authors show that these minor modifications to MH result in improvements in sampling behaviour, as well as the predictive performance of MH.

HMC is the most popular MCMC algorithm within machine learning literature and is used in a wide range of applications including health, finance and cosmology [121, 127, 124, 53, 157, 140, 112, 113, 68, 98, 114]. Its popularity is largely based on its ability to use first-order gradient information of the posterior distribution to guide its exploration, thus

suppressing the random walk behaviour observed with the random walk MH algorithm [124, 17]. This approach can be easily applied to sample from any differentiable target on Euclidean spaces [124, 25]. Betancourt [17] provides the geometrical foundations of HMC, as well as the conditions under which HMC is geometrically ergodic - that is, the algorithm will eventually visit all parts of the parameter space.

Although HMC serves as an improvement on the MH algorithm, it still results in relatively high auto-correlations between the generated samples. This has prompted further development of MCMC algorithms that extend HMC to incorporate second-order gradient information of the posterior [53, 16], techniques that utilise random mass matrices [91] for the auxiliary momentum variable, methods that automatically tune the parameters of HMC [73, 2], algorithms that utilise non-canonical Hamiltonian dynamics [156, 116, 26], approaches that couple HMC chains so as to reduce the variance of the resultant HMC based estimators [136, 112, 113], as well as methods that use modified or shadow Hamiltonians to sample from high dimensional targets [155, 99, 112, 140, 68].

The seminal work of Girolami and Caldehad [53] introduced Riemannian Manifold Hamiltonian Monte Carlo (RMHMC) which improves on HMC by taking into account the local geometry of the target through the use of second-order gradient information. Making use of second-order gradient information is particularly important for ill-conditioned target distributions such as Neal's [123] funnel. The incorporation of local curvature information results in more efficient exploration of the target compared to HMC. A key disadvantage of RMHMC is that the Hamiltonian is not separable, which necessitates the use of an implicit integration scheme that is computationally expensive [53, 35]. Cobb *et al.* [35] introduced an explicit scheme for RMHMC which alleviates the computational burden of using the implicit generalised leapfrog integration scheme without a decrease in sampling performance.

Quantum-Inspired Hamiltonian Monte Carlo (QIHMC) uses a random mass matrix to mimic the behaviour of quantum particles as opposed to the fixed mass of classical particles. This results in better sampling than HMC and RMHMC on spiky and multi-modal distributions [91, 113, 117]. QIHMC achieves the outperformance over HMC without any noticeable increase in the execution time. The main drawback of QIHMC is the requirement for the user to specify the distribution of the mass matrix and the

likely tuning of the parameters of the chosen distribution [91]. Determining the optimal distribution to use for the mass matrix is still an open area of research.

The No-U-Turn Sampler (NUTS) of Hoffman and Gelman [73] automates the tuning of the step size and trajectory length parameters for HMC. The step size parameter is tuned during the burn-in phase using the primal-dual averaging methodology [8] by targeting a user-specified level of acceptance rate in the generated samples [2, 73]. The trajectory length is set by iteratively doubling the trajectory length until specific criteria are met [2, 73, 110, 111]. Empirical results show that NUTS performs at least as efficiently as and sometimes more efficiently than a well-tuned standard HMC method, without requiring user intervention or costly tuning runs [73]. NUTS has also been generalised to Riemannian manifolds [16] and thus allowing the automatic tuning of the parameters of RMHMC. Wang *et al.* [164, 163] introduce a Bayesian optimisation framework for tuning the parameters of HMC and RMHMC samplers. The authors show that their approach is ergodic and, in some instances, precludes the need for more complex samplers. The empirical results show that this approach has a higher effective number of samples per unit of computation used than NUTS. Hoffman *et al.* [71] propose an adaptive MCMC scheme for tuning the trajectory length parameter in HMC and show that this new technique typically yields higher effective samples sizes normalised by the number of gradient evaluation when compared to NUTS. This approach can also be easily run in parallel and use GPUs, unlike NUTS.

Magnetic Hamiltonian Monte Carlo (MHMC) utilises non-canonical dynamics to better explore the posterior [156, 26, 112]. MHMC introduces a magnetic field to HMC in addition to the force field already present HMC. This magnetic field results in faster convergence and lower auto-correlations in the generated samples [59, 156, 107]. When the magnetic component is removed, MHMC has the same performance as HMC. This implies that the magnetic component adds a degree of freedom to improve the performance of HMC. MHMC has been extended to manifolds by Brofos and Lederman [25] and shows good improvement over MHMC. However, MHMC has the disadvantage that the magnetic component has to be specified by the user. In the existing literature on MHMC, there are no automated means of tuning the magnetic component [26, 156, 112]. In addition, the use of a random mass matrix for the momentum variable in MHMC as

well as the automatic tuning of the step size and trajectory length parameters in MHMC is yet to be explored in the literature.

Since the seminal work of Izaguirre and Hampton [77] on integrator-dependent shadow Hamiltonians, there has been a proliferation of shadow HMC methods in the literature. The shadow Hamiltonian methods are premised on the fact that shadow Hamiltonians are better conserved when compared to the true Hamiltonians [77]. This allows one to use larger step sizes or perform sampling on problems with larger dimensions, without a significant decrease in the acceptance rates when compared to HMC methods [4, 140, 68]. To control the momentum generation, the authors introduce a parameter c, which determines how close the true and the shadow Hamiltonians are. The momentum generation increases the overall computational time of the method. In addition, the algorithm requires the user to tune the parameter c to attain optimal results.

Sweet *et al.* [155] improve on the work of Izaguirre and Hampton [77] by using a canonical transformation on the parameters and momentum. This canonical transformation is substituted into the non-separable Hamiltonian introduced in Izaguirre and Hampton [77] so that it now becomes separable. The canonical transformation results in a processed leapfrog integration scheme which is more computationally efficient when compared to the original shadow HMC method of Izaguirre and Hampton [77]. This is because computationally expensive momentum generation for the non-separable Hamiltonian is no longer required. Furthermore, no new hyperparameters are introduced, with the method having the same parameters as traditional HMC. The authors refer to this new method as the Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) algorithm. The tuning of the parameters of S2HMC, and shadow Hamiltonian methods in general, is yet to be explored in the literature.

Partial momentum refreshment has been utilised by Radivojevic and Akhmatskay [140] and Akhmatskaya and Reich [4] to generate momenta in the context of non-separable Hamiltonians. Radivojevic and Akhmatskay [140] also consider higher-order integrators and their corresponding shadow Hamiltonians and propose the Mix and Match Hamiltonian Monte Carlo algorithm, which provides better sampling properties to HMC. Heide *et al.* [68] derive a non-separable shadow Hamiltonian for the generalised leapfrog integrator used in RMHMC, which results in improved performance relative

to sampling from the true Hamiltonian. The authors employed partial momentum refreshment to generate the momenta but do not offer an approach for tuning the partial momentum refreshment parameter that they introduce.

The use of partial momentum refreshment in MHMC and S2HMC is yet to be considered in the literature. Horowitz [74] employed a partial momentum update to HMC and found that it significantly improved the performance of HMC. That is, keeping some of the dynamics of the chain improved performance without harming the legitimacy of the Monte Carlo method [74, 75, 42, 118, 116]. Given that MHMC is closely related to HMC, one would expect that employing partial momentum refreshment to MHMC would result in the same or better sampling properties as observed by Horowitz [74] on HMC. Similarly, one would expect that incorporating partial momentum refreshment within the processed leapfrog integrator in S2HMC could potentially improve the performance of S2HMC.

Although HMC and its variants discussed above serve as an improvement to other MCMC methods, like other MCMC methods, it still suffers from the presence of autocorrelations in the generated samples [53, 89]. This results in the high variance of HMC based estimators. One approach of tackling the high variance of MCMC estimators is by using results from MCMC coupling theory [78, 69]. The coupling of MCMC methods has been used as a theoretical tool to prove convergence behaviour of Markov chains [146, 80, 79, 78, 20]. More recently, MCMC couplings have been studied for HMC with good results [125, 136]. Markov chain coupling has also been used to provide unbiased HMC estimators [55, 78, 69]. Piponi *et al.* [136] use approximate coupling theory to construct the Antithetic Hamiltonian Monte Carlo (A-HMC) algorithm. These anticorrelated chains are created by running the second chain with the auxiliary momentum variable having the opposite sign of the momentum of the first chain [136, 112]. Their results show that adding antithetic sampling to HMC increases the effective sample size rates. Incorporating antithetic sampling in MHMC and S2HMC to reduce their variance is yet to be explored in the literature.

Research objectives: Our primary objective in this thesis is to improve MHMC and S2HMC by developing more efficient and alternative methodologies to enhance their performance in sampling from various probabilistic machine learning models. We aim to

Table 1.1: This table highlights the gaps in the literature that we have identified. A NO entry means that the item is yet to be considered in the literature. A N/A entry means that the item is not applicable to the algorithm. For the items that have already been addressed in the literature, we provide an appropriate reference.

Algorithm	Partial Momentum Retention	Shadow Density	Random Mass Matrix	Adaptive Approaches	Antithetic Sampling
MH	[126]	N/A	[115]	[62, 103]	[136]
MALA	[126]	N/A	NO	[9]	[136]
HMC	[74]	[155, 77]	[91]	[73, 2]	[136]
MHMC	NO	NO	NO	NO	NO
RMHMC	[68]	[68]	NO	[16, 73]	[113]
S2HMC	NO	N/A	NO	NO	NO

do this by combining ideas of utilising random mass matrices for the auxiliary momentum variable, employing partial momentum refreshment, deriving shadow Hamiltonians, automatically tuning the parameters of the methods as well as using antithetic sampling to reduce the variance of MHMC and S2HMC. This combination of ideas is based on addressing the gaps we identified in the literature as summarised in Table 1.1. A secondary aim of this thesis is to apply MCMC methods in areas that have a societal impact within the South African context. Thus, we apply Bayesian inference to analysing South African municipal financial statement audit outcomes to understand the financial state of South African local government entities. In Section 1.2 below, we outline the original contributions to knowledge provided by this thesis in more detail.

1.2 Thesis Contributions

The original contributions of this thesis are both of a theoretical nature in terms of new or improved methods and novel application areas using the Bayesian framework. The contributions of this thesis are summarised in Figure 1.1 and presented in detail below:

1. We start by introducing a new technique which we refer to as the Quantum-Inspired

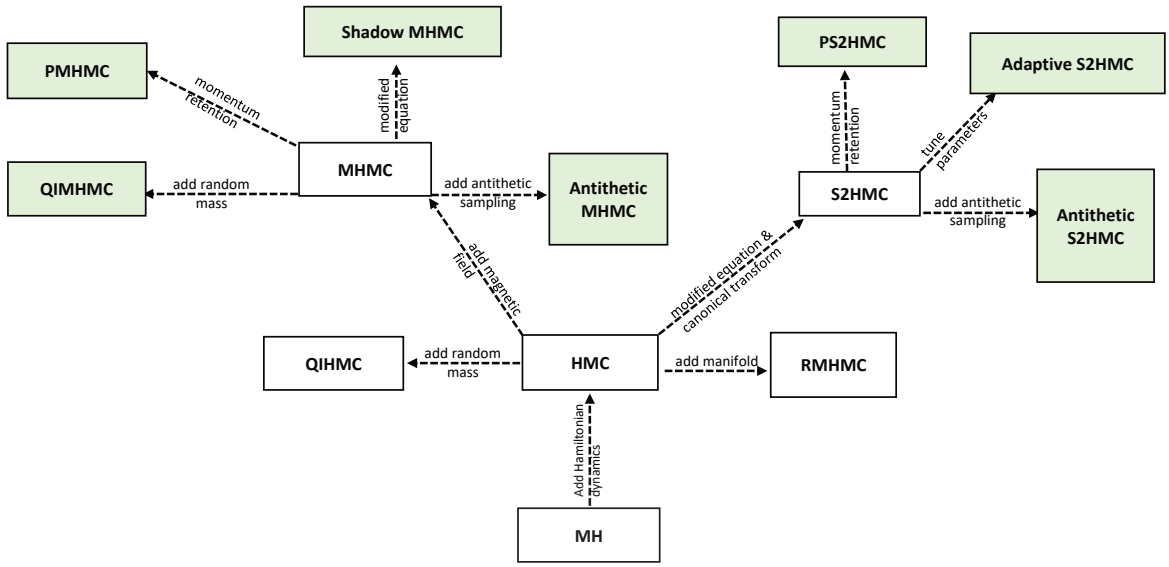


Figure 1.1: The algorithmic contributions of this thesis are highlighted as the green blocks. The arrows indicate the relationship between the different algorithms and show how our proposed algorithms are derived from existing methods.

Magnetic Hamiltonian Monte Carlo (QIMHMC) algorithm. This method uses a random mass matrix instead of a fixed mass for the auxiliary momentum variable in MHMC. This approach mimics the behaviour of quantum particles and leads to better results when compared to HMC, MHMC and QIHMC.

2. We proceed to present novel sampling techniques that employ partial momentum refreshments in the generation of each sample. These are the Magnetic Hamiltonian Monte Carlo With Partial Momentum Refreshment (PMHMC) and Separable Shadow Hamiltonian Hybrid Monte Carlo With Partial Momentum Refreshment (PS2HMC). These two new algorithms illustrate the sampling benefits of not fully refreshing the auxiliary momentum variable in MHMC and S2HMC respectively. Numerical experiments on numerous targets show that PMHMC outperforms HMC, MHMC and Hamiltonian Monte Carlo With Partial Momentum Refreshment (PHMC) while PS2HMC outperforms S2HMC and PMHMC on an Effective Sample Size (ESS) basis.

3. We further derive the fourth-order modified Hamiltonian associated with the numerical integrator employed in MHMC. From this shadow Hamiltonian, we construct a novel sampler which we refer to as Shadow Magnetic Hamiltonian Monte Carlo (SMHMC). Empirical results show that this new method outperforms MHMC across various benchmarks.

4. We then address the issue of automatically tuning the parameters of the S2HMC algorithm. We present the adaptive S2HMC algorithm which extends the NUTS methodology to use the processed leapfrog integrator in S2HMC. This adaptive approach is shown to outperform the appropriately tuned S2HMC method across numerous benchmark problems.

5. We consider the use of antithetic sampling for MHMC and S2HMC to reduce the overall variance of the estimators based on these samplers, or equivalently to increase the effective sample size of these samplers. The results show that the new methods with antithetic sampling outperform the original methods without antithetic sampling.

6. We present the first-in-literature application of a Bayesian approach in modelling audit outcomes of local government entities. Furthermore, we perform Automatic Relevance Determination (ARD) using MCMC methods to identify which financial ratios are essential for modelling audit outcomes of municipalities. Stakeholders could use these results in understanding what drives the audit outcomes of South African municipalities.

1.3 Thesis Scope

The scope of the thesis can be summarised as follows:

- Our primary focus is on samplers that are based on Hamiltonian dynamics, and MHMC and S2HMC in particular

- We only consider Hamiltonian samplers on the Euclidean manifold

- We do not consider any variational inference techniques
- We do not consider other tuning approaches except for those based on the state-of-art NUTS methodology
- We limit our focus to incorporating antithetic sampling to HMC based samplers. Other variance reduction techniques such as control variates are not considered and
- When Bayesian Neural Network (BNN) models are considered, we limit ourselves to Multilayer Perceptron (MLP) architectures with one hidden layer, five hidden units and a single output.

1.4 Publications

The following peer-reviewed journal and conference articles have been produced during this thesis. Furthermore, a book entitled *Hamiltonian Monte Carlo In Machine Learning* based on the work in this thesis has been accepted for publication by ***Elsevier*** in 2022. The Python code used to produce the results in this thesis can be found in the following github repository: github.com/WilsonMongwe/hybridtorch.

1. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Quantum-Inspired Magnetic Hamiltonian Monte Carlo.* Plos One, vol. 16, no. 10, pp. e0258277.
2. **Mongwe, W.T.**, Sigodi, T., Mbuvha, R. and Marwala, T., 2022. *Probabilistic Inference Of South African Equity Option Prices Under Jump-Diffusion Processes.* In 2022 IEEE Computational Intelligence for Financial Engineering and Economics. *To Appear.*
3. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T. 2021. *Locally Scaled and Stochastic Volatility Metropolis–Hastings Algorithms.* Algorithms, vol. 14, no. 12: 351.
4. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Magnetic Hamiltonian Monte Carlo With Partial Momentum Refreshment.* IEEE Access, vol. 9, pp. 108009-108016.

5. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Bayesian Inference of Local Government Audit Outcomes*. Plos One, doi: 10.1371/journal.pone.0261245

6. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Utilising Partial Momentum Refreshment In Separable Shadow Hamiltonian Hybrid Monte Carlo*. IEEE Access, vol. 9, pp. 151235-151244.

7. Mbuvha, R., **Mongwe, W.T.** and Marwala, T. 2021. *Separable Shadow Hamiltonian Hybrid Monte Carlo for Bayesian Neural Network Inference in wind speed forecasting*. Energy and AI (2021): 100108.

8. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Adaptive Magnetic Hamiltonian Monte Carlo*. IEEE Access, vol. 9, pp. 152993-153003.

9. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T. 2021. *On Voter Characterisation in Developing Democracies*. NEURIPS AI for Credible Elections Workshop.[1]

10. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Adaptively Setting the Path Length for Separable Shadow Hamiltonian Hybrid Monte Carlo*. IEEE Access, vol. 9, pp. 138598-138607.

11. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Antithetic Magnetic and Shadow Hamiltonian Monte Carlo*. IEEE Access, vol. 9, pp. 49857-49867.

12. **Mongwe, W.T.** and Malan, K.M., 2020. *A survey of automated financial statement fraud detection with relevance to the South African context*. South African Computer Journal, vol. 32, no. 1, pp. 74-112.

13. **Mongwe, W.T.** and Malan, K.M., 2020. *The Efficacy of Financial Ratios for Fraud Detection Using Self Organising Maps*. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI).pp. 1100-1106. IEEE.

The following manuscript is currently under review at an international journal:

1. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T. *Shadow Magnetic Hamiltonian Monte Carlo*.

[1] https://nips.cc/Conferences/2021/ScheduleMultitrack?event=21857

1.5 Thesis Outline

The remainder of this thesis proceeds as follows:

- **Chapter 2:** In this chapter, we provide a background into MCMC. Furthermore, we review Hamiltonian dynamics-based samplers. We focus on HMC and its variants and provide detailed descriptions of the methods that provide the basis of the new algorithms introduced in this thesis.
- **Chapter 3:** This chapter covers the benchmark problems, real-world application areas, and the performance metrics used throughout this thesis. We also introduce a new dataset into the literature being the South African municipal financial statement audit outcome dataset. We conduct exploratory data analysis on this dataset using a Self Organising Map (SOM) to understand the nature of this dataset better.
- **Chapter 4:** In this chapter, we present the novel QIMHMC algorithm which utilises a random mass matrix for the auxiliary momentum variable in order to enhance the sampling performance of MHMC.
- **Chapter 5:** This chapter presents the PMHMC and PS2HMC algorithms which employ partial momentum refreshment to improve MHMC and S2HMC respectively.
- **Chapter 6:** In this chapter, we introduce the SMHMC algorithm which uses the fourth-order shadow Hamiltonian of the leapfrog-like integrator in MHMC to enhance the sampling performance of MHMC.
- **Chapter 7:** This chapter introduces an adaptive algorithm for S2HMC to tune the step size and trajectory length parameters of S2HMC by extending the NUTS methodology.
- **Chapter 8:** This chapter employs antithetic sampling to reduce the variance of MCMC estimators. We introduce two new algorithms, which are antithetic versions of MHMC and S2HMC.

- **Chapter 9:** This chapter presents the first-in-literature fully Bayesian approach to analysing municipal financial statement audit outcomes. ARD is employed to extract the most critical features for modeling this domain.
- **Chapter 10:** This chapter provides a summary of our original contributions to knowledge as well as ongoing and future work.

Chapter 2

Review of Hamiltonian Samplers

2.1 Introduction

In this chapter, we provide a background to MCMC methods with an emphasis on samplers that utilise Hamiltonian dynamics. These samplers form the basis on which we construct our novel algorithms later in this thesis. This chapter proceeds by first providing an introduction into MCMC and the MH algorithm, after which we study HMC and its various extensions that currently exist in the literature.

2.2 Background to Markov Chain Monte Carlo

Suppose that we are interested in estimating the expectation of some arbitrary function $f(x)$ with $x \in \mathbb{R}^D$ having a well specified probability distribution. Mathematically, this is written as:

$$\mathbb{E}_\mu[f] = \int_{\mathbb{R}^D} f(x) d\mu(x) \tag{2.1}$$

where $f \in L^1(\mu)$ [1] with respect to a probability measure μ, and D is the dimensionality of x. If f and the probability measure μ result in a simple tractable product as an intergrand, one could calculate this expectation analytically. If however, the product of these two functions is not simple, one would have to consider numerical approximations

[1] That is: $\int_{\mathbb{R}^D} |f(x)| d\mu(x) < \infty$.

to the integral in equation (2.1). These techniques could be deterministic quadrature (e.g. Trapezoidal rule [141] and various forms of Gaussian quadrature [93]), Monte Carlo [65], Sparse grids [28] and Bayesian quadrature [131, 60], amongst others.

Deterministic quadrature methods perform poorly as the dimension D of the problem increases. This is due to the convergence rate being bounded as outlined in Bakhalov's theorem [14]. The bulk of quadrature methods were created for integrals with a single dimension. One would then obtain multi-dimensional integrals by employing the tensor product rule from Fubini's theorem [161], which repeats the integrals in each dimension. Multiple one-dimensional integrals result in the function evaluations growing exponentially with the number of dimensions D. The other three methods being Monte Carlo, Sparse grids, and Bayesian quadrature, can (to some extent) overcome this curse of dimensionality.

Sparse grids were developed by Russian mathematician Sergey A. Smolyak and have various applications, including multi-dimensional interpolation and numerical integration [172, 15, 28, 84]. Sparse grids were constructed with a focus on extending full grids methods to higher dimensions, and come at the cost of a slight increase in error [84, 171]. Full-grid approaches are only feasible for small dimensions. For $D > 4$, full grids become computationally expensive as the complexity of full grids grows exponentially with D [15, 171].

Bayesian quadrature is a class of numerical algorithms for solving multi-dimensional integrals in a probabilistic fashion [24, 131]. It has the advantage of providing uncertainty over the solution of the integral. This technique has been successfully utilised to calculate the Bayesian evidence metric [60, 131], which can be used for model comparison and selection. This approach uses a small number of observations of the likelihood to infer a distribution of the integrals akin to those in equation (2.1) using Gaussian processes [60, 131]. Bayesian quadrature offers improved sample efficiency, which is essential for samples drawn from complex and computationally expensive models. This technique has, however, displayed high variance around the integral estimate and convergence diagnostics due to variations in the sample observations selected [60].

Monte Carlo methods utilise repeated random sampling instead of deterministic points as used in deterministic quadrature, and are easier to extend to multidimen-

sional integrals. For this approach, the estimator of the integral in equation (2.1) using n samples has a standard deviation that is proportional to $\frac{1}{\sqrt{n}}$, and importantly does not rely on the dimension D [159]. At the same time, the worst-case error for quadrature is $\propto n^{-\frac{r}{D}}$ where r is the number of quadrature points, with the error depending on D [159]. This suggests that the Monte Carlo approach may perform better when D is large. However, it is not straightforward to directly compare quadrature with Monte Carlo as the former is a frequentist notion while the latter is Bayesian [159].

The Monte Carlo estimate of the integral in equation (2.1) is given as the average of the observations of $x_i \sim \mu$ as [17]:

$$\mathbb{E}_\mu[f] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{2.2}$$

The difficulty in deploying Monte Carlo integration in practice is generating representative samples x_i of the distribution μ.

MCMC techniques, such as the MH algorithm [65] and Slice sampling [123], are a significant subset of Monte Carlo algorithms which can be used to generate $x_i \sim \mu$. In this work, we focus on Monte Carlo based, particularly MCMC, methods for solving the integral in equation (2.1) via the generation of draws from target posterior distributions. MCMC algorithms produce observations of the distribution of interest using a Markov chain [99]. This Markov chain is created so that the equilibrium density is equal to the distribution of interest.

Before delving further into MCMC methods, we recall the following ideas from the theory of Markov Chains taken from [66].

Definition 1. *A Markov chain is a set of stochastic variables* $\{a_1, a_2, ..., a_{n-1}, a_n\}$ *that satisfy:*

$$I\!P(a_n|a_1, a_2, ..., a_{n-1}) = I\!P(a_n|a_{n-1}) \tag{2.3}$$

That is, the density of the current position is independent of the past when given the previous position [66]. This construction implies that one only requires the initial density and a transition density, governing movements between states, to completely specify a Markov chain [99, 66].

Definition 2. *π is the stationary distribution for a Markov chain* $\iff$

$$\pi(b) = \int_a I\!P(b|a)\pi(a)da \quad \forall b, a \tag{2.4}$$

This definition means that the transitions of the chain from one state to the other leave such a distribution invariant [66]. This equation is often called the fixed point equation [153, 66].

Definition 3. *A Markov chain is irreducible and aperiodic* $\iff$ *it is ergodic.*

Irreducibility refers to the chain being able to traverse to any state from any other state [66]. Aperiodic means that the chain eventually returns to the current state after a finite number of steps.

Definition 4. *If* $\exists\ \pi$ *such that:*

$$I\!P(k|w)\pi(w) = I\!P(w|k)\pi(k) \quad \forall k, w \tag{2.5}$$

then the Markov chain is reversible.

Note that equation (2.5) implies that the transitions from-and-to two different states occurs at the same rate when in *equilibrium* [66]. A chain that satisfies equation (2.5) is said to satisfy detailed balance, which is a concept we consistently revisit in this thesis. To guarantee that π is indeed the stationary distribution, we need the chain to be ergodic as defined in Definition 3 [66]. Detailed balance is a more stringent condition than ergodicity, and is not necessary (although it is sufficient) for a chain to converge to the desired equilibrium distribution [66].

MCMC methods were originally developed by physicists such as Ulam, Von Neumann, Fermi, and Metropolis, amongst others in the 1940s [144]. MCMC approaches have since been successfully employed *inter alia* in cosmology, health, energy and finance [98, 99]. The MH method is one of the oldest and foundational MCMC algorithms and forms the core of more complex MCMC techniques which use advanced proposal distributions. Suppose were are interested in determining the posterior distribution of parameters $\mathbf{w}$ from some model M. This posterior distribution could then be used to determine expectations of the form in equation (2.1). The MH method produces

recommended samples using a proposal distribution $T(\mathbf{w}^*|\mathbf{w})$. The next state $\mathbf{w}^*$ is rejected or accepted using the ratio of likelihoods [27, 99]:

$$\mathbb{P}\big(\text{accept } \mathbf{w}^*\big) = \min\left(1, \frac{\pi(\mathbf{w}^*)T(\mathbf{w}|\mathbf{w}^*)}{\pi(\mathbf{w})T(\mathbf{w}^*|\mathbf{w})}\right) \tag{2.6}$$

where $\pi(\mathbf{w})$ is the stationary target density evaluated at $\mathbf{w}$, for some arbitrary $\mathbf{w}$.

Random walk Metropolis is a version of MH that utilises a Gaussian distribution whose mean is the current state as the proposal distribution [99]. The transition density in random walk Metropolis is $\mathcal{N}(\mathbf{w}, \epsilon\boldsymbol{\Sigma})$, where $\epsilon\boldsymbol{\Sigma}$ is the covariance matrix of the proposal distribution [99]. Note that $\boldsymbol{\Sigma}$ is typically set to be the identity matrix in practice, leaving only ϵ to be specified and tuned by the user. When the proposal distribution is symmetric (e.g. when it is Gaussian), it results in $T(\mathbf{w}|\mathbf{w}^*) = T(\mathbf{w}^*|\mathbf{w})$ and leads to equation (2.6) becoming [99]:

$$\mathbb{P}\big(\text{accept } \mathbf{w}^*\big) = \min\left(1, \frac{\pi(\mathbf{w}^*)}{\pi(\mathbf{w})}\right) \tag{2.7}$$

The proposal density in random walk MH usually results in random walk behaviour, which leads to high auto-correlations between the generated samples as well as slow convergence [99]. Algorithm 1 provides the pseudo-code for the MH algorithm, and Theorem 2.2.1 shows why this algorithm converges to the correct stationary distribution.

Algorithm 1 The Metropolis-Hastings Algorithm

Data: $\mathbf{w}_{\text{init}}, \epsilon, N$ and unnormalised target $\pi(\mathbf{w})$

Result: $(\mathbf{w})_{i=0}^{N}$

$\mathbf{w}_0 \leftarrow \mathbf{w}_{\text{init}}$

for $n \leftarrow 1$ **to** N **do**

  $\boldsymbol{\Sigma} = \mathbf{I}$

  $\mathbf{w}^* \sim T$ with $T = \mathcal{N}(\mathbf{w}, \epsilon\boldsymbol{\Sigma})$

  $\mathbf{w}_n \leftarrow \mathbf{w}^*$ with probability: $\alpha(\mathbf{w}^*|\mathbf{w}) = \min\left(1, \frac{\pi(\mathbf{w}^*)T(\mathbf{w}|\mathbf{w}^*)}{\pi(\mathbf{w})T(\mathbf{w}^*|\mathbf{w})}\right)$

end

Theorem 2.2.1. *The MH algorithm in Algorithm 1 leaves the target distribution invariant.*

Proof.

$$\begin{aligned}
\underbrace{\alpha(\mathbf{w}^*|\mathbf{w})T(\mathbf{w}^*|\mathbf{w})}_{\mathbb{P}(\mathbf{w}^*|\mathbf{w})}\pi(\mathbf{w}) &= \min\left(1, \frac{\pi(\mathbf{w}^*)T(\mathbf{w}|\mathbf{w}^*)}{\pi(\mathbf{w})T(\mathbf{w}^*|\mathbf{w})}\right)T(\mathbf{w}^*|\mathbf{w})\pi(\mathbf{w}) \\
&= \min\left(T(\mathbf{w}|\mathbf{w}^*)\pi(\mathbf{w}^*), T(\mathbf{w}^*|\mathbf{w})\pi(\mathbf{w})\right) \\
&= \min\left(1, \frac{\pi(\mathbf{w})T(\mathbf{w}^*|\mathbf{w})}{\pi(\mathbf{w}^*)T(\mathbf{w}|\mathbf{w}^*)}\right)T(\mathbf{w}|\mathbf{w}^*)\pi(\mathbf{w}^*) \\
&= \underbrace{\alpha(\mathbf{w}|\mathbf{w}^*)T(\mathbf{w}|\mathbf{w}^*)}_{\mathbb{P}(\mathbf{w}|\mathbf{w}^*)}\pi(\mathbf{w}^*) \\
\implies \mathbb{P}(\mathbf{w}^*|\mathbf{w})\pi(\mathbf{w}) &= \mathbb{P}(\mathbf{w}|\mathbf{w}^*)\pi(\mathbf{w}^*)
\end{aligned} \tag{2.8}$$

which is the required result. □

A significant drawback of the MH seminal method is the high auto-correlations between the generated samples. The high auto-correlations lead to slow convergence in the Monte Carlo estimates and consequently the necessity to generate large sample sizes. Approaches that reduce the random walk behavior are those that enhance the classical MH algorithm and utilise more information about the target posterior distributions [53]. The most common extension is to incorporate first-order gradient information to guide the exploration of the target, as well as second-order gradient information to consider the local curvature of the posterior [53]. Methods that use first-order gradient information include methods that utilise Langevin and Hamiltonian dynamics on a Euclidean manifold, while approaches that use second-order gradient information are methods on Riemannian and other manifolds [53, 25, 41, 156, 25]. Methods that use Hamiltonian dynamics on Euclidean spaces have shown great success in practice, and are a focus of this thesis. We assess different aspects of Hamiltonian dynamics-based samplers with a focus on addressing the gaps in the literature that we outlined in Table 1.1.

The remainder of this chapter outlines the methods that form the basis of the new algorithms presented in this thesis. The chapter proceeds as follows: we first review MALA, HMC, MHMC and QIHMC, we then proceed to discuss in Section 2.7 how to improve these samplers via modified or shadow Hamiltonians, we then discuss how one can automatically tune the parameters of these samplers in Section 2.8, and finally

in Section 2.9, we conduct a review of coupling theory which will be employed when reducing the variance of Hamiltonian chains using antithetic sampling.

2.3 Metropolis Adjusted Langevin Algorithm

The MALA is a MCMC method that incorporates first-order gradient information of the target posterior to enhance the sampling behaviour of the MH algorithm [53, 59]. MALA reduces the random walk behaviour of MH via the use of Langevin dynamics which are given as follows [53, 59]:

$$d\mathbf{w}_t = \frac{1}{2}\nabla_{\mathbf{w}} \ln \pi(\mathbf{w})dt + dZ_t \tag{2.9}$$

where $\pi(\mathbf{w})$ represents the unnormalised target distribution (which is the negative log-likelihood), $\mathbf{w}$ is the position vector and Z_t is a Brownian motion process at time t. As the Langevin dynamics are in the form of a stochastic differential equation, we typically will not be able to solve it analytically, and we need to use a numerical integration scheme. The first-order Euler-Maruyama integration scheme is the commonly used integration scheme for solving Langevin dynamics and the update equation is given as: [53, 59]:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \frac{\epsilon^2}{2}\nabla_{\mathbf{w}} \ln \pi(\mathbf{w}) + \epsilon z_t \tag{2.10}$$

where ϵ is the integration step size and $z_t \sim \mathcal{N}(0, \mathbf{I})$.

The Euler-Maruyama integration scheme does not provide an exact solution to the Langevin dynamics, and hence produces numerical integration errors. These errors result in detailed balance being broken, and one ends up not sampling from the correct distribution. In order to ensure detailed balance, an MH acceptance step is required. The transition probability density function of MALA is given as [59]:

$$\begin{aligned} K(\mathbf{w}'|\mathbf{w}) &= \mathcal{N}(\gamma(\mathbf{w}), \epsilon^2\mathbf{I}), \\ K(\mathbf{w}|\mathbf{w}') &= \mathcal{N}(\gamma(\mathbf{w}'), \epsilon^2\mathbf{I}), \\ \gamma(\mathbf{w}') &= \mathbf{w} + \frac{\epsilon^2}{2}\nabla_{\mathbf{w}} \ln \pi(\mathbf{w}), \\ \gamma(\mathbf{w}) &= \mathbf{w}' + \frac{\epsilon^2}{2}\nabla_{\mathbf{w}} \ln \pi(\mathbf{w}'). \end{aligned} \tag{2.11}$$

where $K(\mathbf{w}'|\mathbf{w})$ and $K(\mathbf{w}|\mathbf{w}')$ are transition probability distributions, $\mathbf{w}$ is the current state and $\mathbf{w}'$ is the new proposed state. The acceptance rate of the MALA takes the form:

$$\min\left[1, \frac{\pi(\mathbf{w}')K(\mathbf{w}'|\mathbf{w})}{\pi(\mathbf{w})K(\mathbf{w}|\mathbf{w}')}\right] \tag{2.12}$$

It can be shown that MALA converges in the correct stationary by showing that it satisfies the Fokker–Planck equation as in [53], or by following a similar approach to the proof for MH in Theorem 2.2.1.

Unlike the MH algorithm, the MALA considers the first-order gradient data of the density, which results in the algorithm converging to the stationary distribution at an accelerated rate [53, 59]. However, the generated samples are still highly correlated. Girolami and Caldehad [53] extend MALA from being on a Euclidean manifold to a Riemannian manifold, and hence incorporating second-order gradient information of the target. This approach showed considerable improvements on MALA, with an associated increase in compute time.

In the following sections, we present Hamiltonian dynamics-based methods that explore the posterior distribution more efficiently than MALA.

2.4 Hamiltonian Monte Carlo

The HMC algorithm is composed of two steps: 1) the molecular dynamics step and 2) the Monte Carlo step. The molecular dynamics step involves integrating Hamiltonian dynamics, while the Monte Carlo step employs the MH algorithm to account for any errors introduced by the numerical integrator used in the molecular dynamics step [41, 77, 4, 122, 65, 121, 127]. Note that if we could exactly solve the molecular dynamics step, we would not need the Monte Carlo step.

HMC improves upon the MH [65] algorithm by utilising first-order gradient information of the unnormalised target posterior. This gradient information is used to guide HMC's exploration of the parameter space [41, 127]. This necessities that the target posterior function is differentiable and has support almost everywhere on $\mathbb{R}^D$, which is the case for the majority of machine learning models of interest. In HMC, the position vector $\mathbf{w}$ is augmented with auxiliary momentum variable $\mathbf{p}$, which is typically chosen

to be independent of $\mathbf{w}$. The Hamiltonian $\mathrm{H}(\mathbf{w}, \mathbf{p})$, which represents the total energy, from this system is written as follows:

$$\mathrm{H}(\mathbf{w}, \mathbf{p}) = \mathrm{U}(\mathbf{w}) + \mathrm{K}(\mathbf{p}) \tag{2.13}$$

where $\mathrm{U}(\mathbf{w})$ is potential energy or the negative log-likelihood of the target posterior distribution and $\mathrm{K}(\mathbf{p})$ is the kinetic energy defined by the kernel of a Gaussian with a covariance matrix $\mathbf{M}$ [124]:

$$\mathrm{K}(\mathbf{p}) = \frac{1}{2}\log\left((2\pi)^D|\mathbf{M}|\right) + \frac{\mathbf{p}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{p}}{2}. \tag{2.14}$$

Within this framework, the evolution of the physical system is governed by Hamiltonian dynamics [153, 124]. As the particle moves through time using Hamiltonian dynamics, the total energy is conserved over the entire trajectory of the particle, with kinetic energy being exchanged for potential energy and vice versa to ensure that the Hamiltonian or total energy is conserved [153, 124]. As an illustration, consider a person skating from position A to C as displayed in Figure 2.1. At position A, the person only has potential energy and no kinetic energy, and they only have kinetic energy at point B. At position C, they have both kinetic and potential energy. Throughout the movement from A to C, the total energy will be conserved if the individual traverses the space using Hamiltonian dynamics. This allows the individual to traverse long distances. This energy conservation property of Hamiltonian dynamics is key to the efficiency of HMC in exploring the target posterior.

The equations governing the Hamiltonian dynamics are defined by Hamilton's equations in a fictitious time t as follows [121]:

$$\frac{\mathrm{d}\mathbf{w}}{\partial t} = \frac{\partial \mathrm{H}(\mathbf{w}, \mathbf{p})}{\partial \mathbf{p}}; \quad \frac{\mathrm{d}\mathbf{p}}{\partial t} = -\frac{\partial \mathrm{H}(\mathbf{w}, \mathbf{p})}{\partial \mathbf{w}}. \tag{2.15}$$

which can also be re-expressed as:

$$\frac{d}{\partial t}\begin{bmatrix}\mathbf{w}\\ \mathbf{p}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\nabla_w \mathrm{H(w,\ p)}\\ \nabla_p \mathrm{H(w,\ p)}\end{bmatrix} \tag{2.16}$$

The Hamiltonian dynamics satisfy the following important properties, which make it ideal for efficiently generating distant proposals [17, 122]:

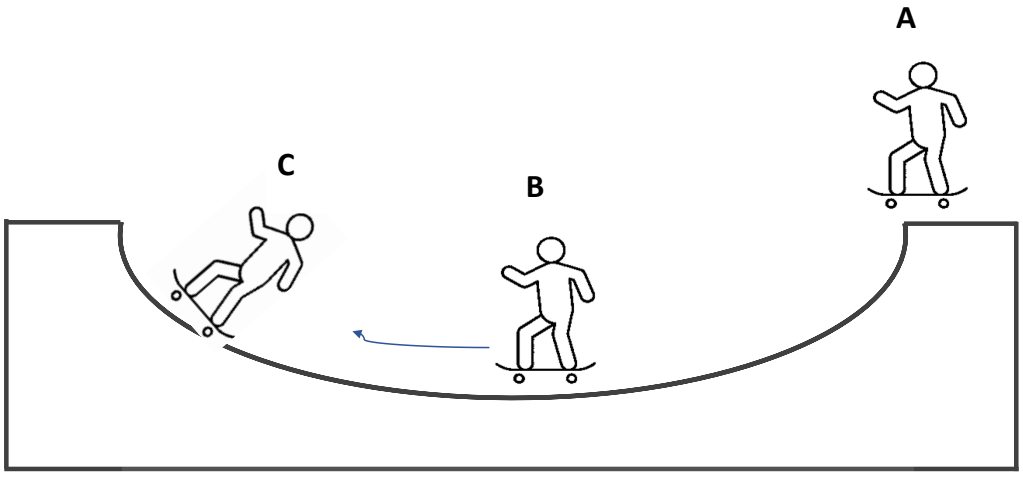


Figure 2.1: Illustration of the conservation of the Hamiltonian (i.e., total energy) through time as the skater moves from position A to C.

1. **Conservation of energy:** That is, the change of the Hamiltonian through time is zero as illustrated in Figure 2.1. Mathematically:

$$\begin{aligned}\frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial t} &= \frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial t} \\ &= \frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial \mathbf{w}}\left(\frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial \mathbf{p}}\right) + \frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial \mathbf{p}}\left(-\frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial \mathbf{w}}\right) \qquad (2.17) \\ \implies \frac{\partial \mathrm{H}(\mathbf{w},\mathbf{p})}{\partial t} &= 0\end{aligned}$$

2. **Reversibility:** That is, the dynamics can be moved forward in time by a certain amount and backwards in time by the same amount to get back to the original position. Mathematically: Let $\Phi_{t,H}\begin{bmatrix}\mathbf{w_0}\\ \mathbf{p_0}\end{bmatrix}$ be the unique solution at time t of equation (2.15) with initial position $\begin{bmatrix}\mathbf{w_0}\\ \mathbf{p_0}\end{bmatrix}$. As the Hamiltonian in equation (2.13) is time-homogeneous, we have that:

$$\begin{aligned}\Phi_{t,H} \circ \Phi_{s,H}\begin{bmatrix}\mathbf{w_0}\\ \mathbf{p_0}\end{bmatrix} &= \Phi_{t+s,H}\begin{bmatrix}\mathbf{w_0}\\ \mathbf{p_0}\end{bmatrix} \\ \implies \Phi_{-t,H} \circ \Phi_{t,H}\begin{bmatrix}\mathbf{w_0}\\ \mathbf{p_0}\end{bmatrix} &= \begin{bmatrix}\mathbf{w_0}\\ \mathbf{p_0}\end{bmatrix}\end{aligned} \qquad (2.18)$$

3. **Volume preservation:** This property serves to simplify the MH step in HMC so that it does not require a Jacobian term, as volume preservation means that the Jacobian term is equal to one [2, 124]. There have also been extensions of HMC that do no not preserve volume [153].

These three properties are significant in that conservation of energy allows one to determine if the approximated trajectory is diverging from the expected dynamics, reversibility of the Hamiltonian dynamics ensures reversibility of the sampler, and volume preservation simplifies the MH acceptance step [68, 124].

The differential equation in equation (2.15) and (2.16) cannot be solved analytically in most instances. This necessitates the use of a numerical integration scheme. As the Hamiltonian in equation (2.13) is separable, to traverse the space, we can employ the leapfrog integrator [41, 121]. The position and momentum update equations for the leapfrog integration scheme are:

$$\begin{aligned}
\mathbf{p}_{t+\frac{\epsilon}{2}} &= \mathbf{p}_t + \frac{\epsilon}{2}\frac{\partial H\left(\mathbf{w}_t, \mathbf{p}_t\right)}{\partial \mathbf{w}} \\
\mathbf{w}_{t+\epsilon} &= \mathbf{w}_t + \epsilon \mathbf{M}^{-1}\mathbf{p}_{t+\frac{\epsilon}{2}} \\
\mathbf{p}_{t+\epsilon} &= \mathbf{p}_{t+\frac{\epsilon}{2}} + \frac{\epsilon}{2}\frac{\partial H\left(\mathbf{w}_{t+\epsilon}, \mathbf{p}_{t+\frac{\epsilon}{2}}\right)}{\partial \mathbf{w}}.
\end{aligned} \tag{2.19}$$

Algorithm 2 Hamiltonian Monte Carlo Algorithm

Input: N, ϵ, L, w_{init}, $H(w, p)$

Output: $(w)_{m=0}^{N}$

$w_0 \leftarrow w_{\text{init}}$
for $m \rightarrow 1$ **to** N **do**
  $p_{m-1} \sim \mathcal{N}(0, \mathbf{M})$ $\quad \leftarrow$ **momentum refreshment**
  $p_m,\ w_m =$ **Leapfrog**$(p_{m-1},\ w_{m-1},\ \epsilon,\ L,\ H)$ in equation (2.19)
  $\delta H = H(w_{m-1}, p_{m-1}) - H(w_m, p_m)$
  $\alpha_m = \min\left(1, \exp\left(\delta H\right)\right)$
  $u_m \sim \text{Unif}(0, 1)$
  $w_m =$ **Metropolis**$(\alpha_m,\ u_m,\ w_m,\ w_{m-1})$ in equation (2.20)
end for

Due to the discretisation errors arising from the numerical integration, the Monte Carlo step in HMC utilises the MH algorithm in which the parameters $\mathbf{w}^*$ proposed by the molecular dynamics step are accepted with probability:

$$\mathrm{P}(\text{accept } \mathbf{w}^*) = \min\left(1, \frac{\exp\left(-\mathrm{H}(\mathbf{w}^*, \mathbf{p}^*)\right)}{\exp\left(-\mathrm{H}(\mathbf{w}, \mathbf{p})\right)}\right). \tag{2.20}$$

Algorithm 2 shows the pseudo-code for the HMC where ϵ is the discretisation step size and L is the trajectory length. The overall HMC sampling process follows a Gibbs sampling scheme, where we *fully* sample the momentum (see line 3 in Algorithm 2) and then sample a new set of parameters given the drawn momentum.

It can be shown that MALA is a special case of HMC with $L = 1$ [53]. Although HMC improves on MH and MALA, it still produces relatively high correlated samples [156, 77]. In the following sections we consider methods that improve on the sampling efficiency of HMC in various ways.

2.5 Magnetic Hamiltonian Monte Carlo

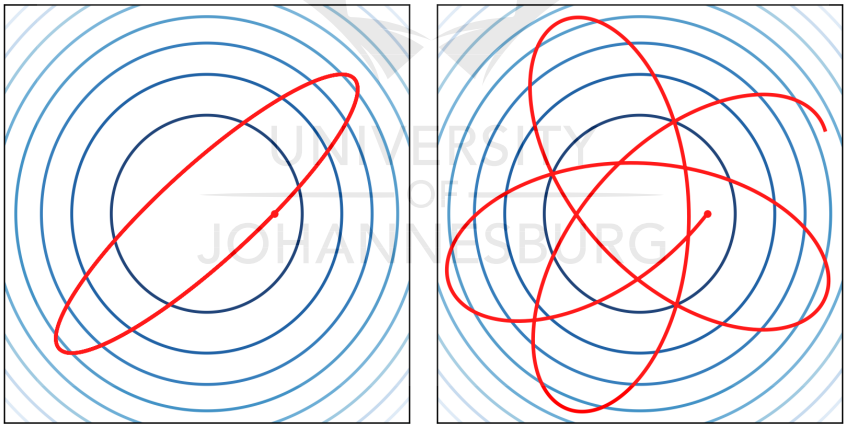


Figure 2.2: Illustration of the sample paths for HMC (on the left) and MHMC (on the right) for a Gaussian distribution with a diagonal covariance matrix. This illustration is taken from Tripuraneni *et al.* [156].

MHMC is a special case of non-canonical HMC using a symplectic structure corresponding to motion of a particle in a magnetic field [156, 26]. MHMC extends HMC by

endowing it with a magnetic field, which results in non-canonical Hamiltonian dynamics [156]. This magnetic field offers a significant amount of flexibility over HMC and encourages more efficient exploration of the posterior, which results in faster convergence and lower auto-correlations in the generated samples [156, 59, 112]. MHMC uses the same Hamiltonian as in HMC, but exploits non-canonical Hamiltonian dynamics where the canonical matrix now has a non-zero element on the diagonal. The MHMC dynamics are given as:

$$\frac{d}{\partial t}\begin{bmatrix}\mathbf{w}\\ \mathbf{p}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{G}\end{bmatrix}\begin{bmatrix}\nabla_w \mathrm{H(w,\ p)}\\ \nabla_p \mathrm{H(w,\ p)}\end{bmatrix} \tag{2.21}$$

where $\mathbf{G}$ is a skew-symmetric [2] (or antisymmetric) matrix and is the term that represents the magnetic field. This also shows that MHMC only differs from HMC dynamics in equation (2.16) by $\mathbf{G}$ being non-zero. When $\mathbf{G} = \mathbf{0}$, MHMC and HMC have the same dynamics. Tripuraneni *et al.* [156] prove that in three dimensions, these dynamics are Newtonian mechanics of a charged particle in a magnetic field. How this magnetic field relates to the force field (e.g. are they orthogonal?) will determine the extent of the sampling efficiency of MHMC over HMC [59]. Figure 2.2 illustrates the vastly different motions that can be generated by MHMC when compared to HMC on an isotropic Gaussian distribution [156].

As with HMC, these non-canonical dynamics cannot be integrated exactly, and we resort to a numerical integration scheme with a MH acceptance step to ensure detailed balance. The update equations for the leapfrog-like integration scheme for MHMC, for the case where $\mathbf{M} = \mathbf{I}$, are given as [156]:

$$\begin{aligned}
\mathbf{p}_{t+\frac{\epsilon}{2}} &= \mathbf{p}_t + \frac{\epsilon}{2}\frac{\partial H\left(\mathbf{w}_t, \mathbf{p}_t\right)}{\partial \mathbf{w}}\\
\mathbf{w}_{t+\epsilon} &= \mathbf{w}_t + \mathbf{G}^{-1}\left(\exp\left(\mathbf{G}\epsilon\right) - \mathbf{I}\right)\mathbf{p}_{t+\frac{\epsilon}{2}}\\
\mathbf{p}_{t+\frac{\epsilon}{2}} &= \exp\left(\mathbf{G}\epsilon\right)\mathbf{p}_{t+\frac{\epsilon}{2}}\\
\mathbf{p}_{t+\epsilon} &= \mathbf{p}_{t+\frac{\epsilon}{2}} + \frac{\epsilon}{2}\frac{\partial H\left(\mathbf{w}_{t+\epsilon}, \mathbf{p}_{t+\frac{\epsilon}{2}}\right)}{\partial \mathbf{w}}.
\end{aligned} \tag{2.22}$$

The above equations show that we can retrieve the update equations of traditional HMC by first performing a Taylor matrix expansion for the exponential and then sub-

[2] That is: $\mathbf{G}^T = -\mathbf{G}$

stituting $\mathbf{G} = 0$. The pseudo-code for the MHMC algorithm is shown in Algorithm 3. It is important to note that we need to flip the sign of $\mathbf{G}$ (see lines 8-15 in Algorithm 3), as we do the sign of $\mathbf{p}$ in HMC, so as to render the MHMC algorithm reversible. In this sense, we treat $\mathbf{G}$ as being an auxiliary variable in the same fashion as $\mathbf{p}$ [156]. In this setup, $\mathbf{p}$ would be Gaussian while $\mathbf{G}$ would have a binary distribution [156] and only taking on the values $\pm\mathbf{G_0}$, with $\mathbf{G_0}$ being specified by the user. Exploring more complex distributions for $\mathbf{G}$ is still an open area of research.

Although MHMC requires matrix exponentiation and inversion as shown in equation (2.22), this only needs to be computed once upfront and stored [156]. Following this approach results in computation time that is comparable to HMC, which becomes more important in models that have many parameters such as neural networks.

As $\mathbf{G}$ only needs to be antisymmetric, there is no guarantee that it will be invertible. In this case, we need first to diagonalise $\mathbf{G}$ and separate its invertible or singular components [156]. As $\mathbf{G}$ is strictly antisymmetric, we can express it as $i\mathbf{H}$ where $\mathbf{H}$ is a Hermitian matrix, and can thus be diagonilised over the space of complex numbers $\mathbb{C}$ as [156]:

$$\mathbf{G} = \begin{bmatrix} W_{\mathbf{\Lambda}} & W_{\mathbf{0}} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} W_{\mathbf{\Lambda}}^T \\ W_{\mathbf{0}}^T \end{bmatrix} \tag{2.23}$$

where $\mathbf{\Lambda}$ is a diagonal submatrix consisting of the nonzero eigenvalues of $\mathbf{G}$, columns of $W_{\mathbf{\Lambda}}$, and $W_{\mathbf{0}}$ are the eigenvectors of $\mathbf{G}$ corresponding to its nonzero and zero eigenvalues, respectively. This leads to the following update for $\mathbf{w}$ in equation (2.22) [156]:

$$\mathbf{w}_{t+\epsilon} = \mathbf{w}_t + \begin{bmatrix} W_{\mathbf{\Lambda}} & W_{\mathbf{0}} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}^{-1}\left(\exp(\mathbf{\Lambda} t) - \mathbf{I}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} W_{\mathbf{\Lambda}}^T \\ W_{\mathbf{0}}^T \end{bmatrix} \mathbf{p}_{t+\frac{\epsilon}{2}} \tag{2.24}$$

It is worthwhile noting that when $\mathbf{G} = 0$ in equation (2.24) then the flow map will reduce to an Euler translation as in traditional HMC [156].

2.6 Quantum-Inspired Hamiltonian Monte Carlo

Inspired by the energy-time uncertainty relation from quantum mechanics, Liu and Zhang [91] developed the QIHMC algorithm which sets $\mathbf{M}$ to be random with a probability distribution rather than a fixed mass as is the case in HMC [127]. That is, QIHMC

sets the covariance mass matrix $\mathbf{M}$ in HMC in equation (2.14) to be a stochastic process [91]. The authors show that QIHMC outperforms HMC, RMHMC and the NUTS [73] on a variety of spiky and multi-modal distributions which occur in sparse modeling via bridge regression, image denoising and BNN pruning, and furthermore leaves the target distribution invariant. The proof that QIHMC leaves the target distribution invariant utilises Bayes theorem to marginalize out the random mass matrix, and can be found in Liu and Zang [91].

Algorithm 3 Magnetic Hamiltonian Monte Carlo Algorithm

Input: N, ϵ, L, w_{init}, $H(w,p)$, G

Output: $(w)_{m=0}^{N}$

$w_0 \leftarrow w_{\text{init}}$
for $m \rightarrow 1$ **to** N **do**
  $p_{m-1} \sim \mathcal{N}(0, \mathbf{M})$ $\leftarrow$ **momentum refreshment**
  $p_m,\ w_m =$ **Integrator**$(p_{m-1},\ w_{m-1},\ \epsilon,\ L,\ \mathrm{G},\ H)$ in equation (2.22)
  $\delta H = H(w_{m-1}, p_{m-1}) - H(w_m, p_m)$
  $\alpha_m = \min\left(1, \exp\left(\delta H\right)\right)$
  $u_m \sim \mathrm{Unif}(0,1)$
  if $\alpha_m > u_m$ **then**
    $w_m = w_m$
    $G = -G,\ p_m = -p_m$
  else
    $w_m = w_{m-1}$
  end if
  $p_m = -p_m \leftarrow$ **flip momentum**
  $G = -G \leftarrow$ **flip magnetic field**
end for

A key feature of QIHMC is that it requires very little modification to existing HMC implementations, which makes it straightforward to incorporate into already existing HMC code. HMC is typically inefficient in sampling from spiky and multi-modal distributions [91]. Setting the mass matrix $\mathbf{M}$ to be random alleviates this drawback of

HMC.

Suppose it is straightforward to generate samples from the distribution of the mass matrix $\mathbf{M} \sim \mathcal{P}_{\mathbf{M}}$. In that case, there will be very little additional computational overhead in QIHMC, but with potentially significant improvements in target exploration [91]. It is worth noting that HMC is a special case of QIHMC when the probability distribution over the mass matrix is the Dirac delta function [91]. This feature illustrates how closely related these two algorithms are.

As the Hamiltonian in QIHMC is the same as in HMC, we integrate the dynamics using the leapfrog scheme outline in equation (2.19). The only difference with HMC is that the mass matrix of the auxilary momentum variable is chosen from a user-specified distribution for each generation of a new sample. The pseudo-code for the QIHMC algorithm is shown in Algorithm 4. Line 3 in Algorithm 4 is the only difference between QIHMC and HMC.

Algorithm 4 Quantum-Inspired Hamiltonian Monte Carlo Algorithm

Input: N, ϵ, L, w_{init}, $H(w, p)$

Output: $(w)_{m=0}^{N}$

1: $w_0 \leftarrow w_{\text{init}}$
2: **for** $m \rightarrow 1$ **to** N **do**
3: $\quad \mathbf{M} \sim \mathcal{P}_{\mathbf{M}}(\mathbf{M})$ $\quad \leftarrow$ **only difference with HMC** in Algorithm 2
4: $\quad p_{m-1} \sim \mathcal{N}(0, \mathbf{M})$ $\quad \leftarrow$ **momentum refreshment**
5: $\quad p_m,\ w_m =$ **Leapfrog**$(p_{m-1},\ w_{m-1},\ \epsilon,\ L,\ H)$ in equation (2.19)
6: $\quad \delta H = H(w_{m-1}, p_{m-1}) - H(w_m, p_m)$
7: $\quad \alpha_m = \min\left(1, \exp\left(\delta H\right)\right)$
8: $\quad u_m \sim \text{Unif}(0, 1)$
9: $\quad w_m =$ **Metropolis**$(\alpha_m,\ u_m,\ w_m,\ w_{m-1})$ in equation (2.20)
10: **end for**

A fundamental limitation of QIHMC is what probability distribution $\mathcal{P}_{\mathbf{M}}$ over $\mathbf{M}$ to use, as well as the potential tuning of the parameters of the chosen probability distribution. An improperly chosen distribution could result in wasted computation without any significant improvements to sampling performance. Liu and Zhang [91] employ a

log-normal distribution for the diagonal elements of the mass matrix, but do not provide a mechanism or a heuristic to tune the parameters to obtain optimal results. The selection of the optimal $\mathcal{P}_{\mathbf{M}}$ is still an open research problem.

2.7 Separable Shadow Hamiltonian Hybrid Monte Carlo

The leapfrog integrator for HMC only preserves the Hamiltonian, that is, the total energy of the system, up to second order $\mathcal{O}(\epsilon^2)$ [77, 155, 155]. This leads to a larger than expected value for δH in line 5 of Algorithm 2 for long trajectories, which results in more rejections in the MH step in line 8 of Algorithm 2. To increase the accuracy of the preservation of the total energy to higher orders, and consequently maintain high acceptance rates, one could: 1) decrease the step size and thus only consider short trajectories, or 2) utilise numerical integration schemes which preserve the Hamiltonian to a higher order, 3) or a combination of 1) and 2). These three approaches typically lead to a high computational burden, which is not ideal [68, 140].

An alternative strategy is to assess the error produced by feeding the solution backward through [63, 99] the leapfrog integration scheme in equation (2.19), to derive a modified Hamiltonian whose energy is preserved to a higher-order by the integration scheme than the true Hamiltonian [99]. This modified Hamiltonian is also referred to as the shadow Hamiltonian. We then sample from the shadow density and correct for the induced bias via importance sampling as is done in [68, 155, 77, 140], among others.

Shadow Hamiltonians are perturbations of the Hamiltonian that are by design exactly conserved by the numerical integrator [140, 77, 112, 99]. In the case of shadow Hamiltonian Hybrid Monte Carlo, we sample from the importance distribution defined by the shadow Hamiltonian [99]:

$$\hat{\pi} \propto \exp\big(-\tilde{\mathrm{H}}^{[k]}(\mathbf{w}, \mathbf{p})\big) \tag{2.25}$$

where $\tilde{\mathrm{H}}^{[k]}$ is the shadow Hamiltonian defined using backward error analysis of the numerical integrator up to the k^{th} order [99]. These modified densities can be proved to be Hamiltonian for symplectic integrators such as the leapfrog integrator [155, 77, 99, 112].

In this thesis, we focus on a fourth-order truncation of the shadow Hamiltonian under the leapfrog integrator. Since the leapfrog is second-order accurate ($\mathcal{O}^2$), the fourth-order truncation is conserved with higher accuracy ($\mathcal{O}^4$) than the true Hamiltonian, which should improve the acceptance rate. In Theorem 2.7.1, we derive the fourth-order shadow Hamiltonian corresponding to the leapfrog integrator.

Theorem 2.7.1. *Let $H : R^d \times R^d = R$ be a smooth Hamiltonian function. The fourth–order shadow Hamiltonian function $\hat{\mathrm{H}} : R^d \times R^d = R$ corresponding to the leapfrog integrator of HMC is given by:*

$$\hat{\mathrm{H}}(\mathbf{w}, \mathbf{p}) = \mathrm{H}(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12}\left[K_{\mathbf{p}} U_{\mathbf{ww}} K_{\mathbf{p}}\right] - \frac{\epsilon^2}{24}\left[U_{\mathbf{w}} K_{\mathbf{pp}} U_{\mathbf{w}}\right] + \mathcal{O}(\epsilon^4) \tag{2.26}$$

Proof. The Hamiltonian vector field: $\overrightarrow{H} = \overrightarrow{A} + \overrightarrow{B}$ will generate the exact flow corresponding to exactly simulating the HMC dynamics [156]. We obtain the shadow density by making use of the separability of the Hamiltonian in equation (2.13). The leapfrog integration scheme in equation (2.19) splits the Hamiltonian as:

$$H(\mathbf{w}, \mathbf{p}) = H_1(\mathbf{w}) + H_2(\mathbf{p}) + H_1(\mathbf{w}) \tag{2.27}$$

and exactly integrates each sub-Hamiltonian. Through the Baker-Campbell-Hausdorff (BCH) formula, we obtain [64]:

$$\begin{aligned} \Phi_{\epsilon,H} &= \Phi_{\epsilon,H_1(\mathbf{w})} \circ \Phi_{\epsilon,H_2(\mathbf{p})} \circ \Phi_{\epsilon,H_1(\mathbf{w})} \\ &= \exp\left(\frac{\epsilon}{2}\overrightarrow{B}\right) \circ \exp\left(\epsilon\overrightarrow{A}\right) \circ \exp\left(\frac{\epsilon}{2}\overrightarrow{B}\right) \\ &= \mathrm{H}(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12}\{K, \{K, U\}\} - \frac{\epsilon^2}{24}\{U, \{U, K\}\} + \mathcal{O}(\epsilon^4) \end{aligned} \tag{2.28}$$

where the canonical Poisson brackets are defined as:

$$\begin{aligned} \{f, g\} &= \left[\nabla_{\mathbf{w}} f, \nabla_{\mathbf{p}} f\right] \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \left[\nabla_{\mathbf{w}} g, \nabla_{\mathbf{p}} g\right]^T \\ &= -\nabla_{\mathbf{p}} f \nabla_{\mathbf{w}} g + \nabla_{\mathbf{w}} f \nabla_{\mathbf{p}} g \end{aligned} \tag{2.29}$$

The shadow Hamiltonian for the leapfrog integrator is then:

$$\hat{\mathrm{H}}(\mathbf{w}, \mathbf{p}) = \mathrm{H}(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12}\left[K_{\mathbf{p}} U_{\mathbf{ww}} K_{\mathbf{p}}\right] - \frac{\epsilon^2}{24}\left[U_{\mathbf{w}} K_{\mathbf{pp}} U_{\mathbf{w}}\right] + \mathcal{O}(\epsilon^4) \tag{2.30}$$

It is worth noting that the shadow Hamiltonian in (2.30) is conserved to fourth-order [68, 155, 140]. □

Izaguirre and Hampton [77] utilised the shadow Hamiltonian in Theorem 2.7.1 to derive the Shadow Hamiltonian Monte Carlo (SHMC) algorithm, which produce better efficiency and acceptance rates when compared to HMC. It is worth noting that the shadow Hamiltonian in Theorem 2.7.1 is not separable, i.e., the terms that depend on $\mathbf{w}$ and $\mathbf{p}$ can not be separated. The non-separable shadow Hamiltonian is not ideal as it requires computationally expensive methods for generating the momenta. Sweet *et al.* improve on the SHMC method by introducing the S2HMC algorithm which utilises a processed leapfrog integrator to create a separable Hamiltonian [155, 99, 112]. The separable Hamiltonian in S2HMC is as follows:

$$\tilde{\mathrm{H}}(\mathbf{w},\mathbf{p}) = \mathrm{U}(\mathbf{w}) + \mathrm{K}(\mathbf{p}) + \frac{\epsilon^2}{24}\mathrm{U}_{\mathbf{w}}^{T}\mathrm{M}^{-1}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^4) \tag{2.31}$$

which is obtained by substituting a canonical transformation $(\hat{\mathbf{w}},\hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w},\mathbf{p})$ into (2.30). We provide a detailed derivation of the separable Hamiltonian in equation (2.31) in Appendix B.

The canonical transformation required to generate the separable Hamiltonian in S2HMC is a transformation of both the position $\mathbf{w}$ and the momenta $\mathbf{p}$ variables so as to remove the mixed terms in the shadow Hamiltonian in Theorem 2.7.1. This transformation or map should commute with reversal of momenta and should preserve phase space volume so that the resulting S2HMC algorithm satisfies detailed balance [155, 99, 112]. Propagation of positions and momenta on this shadow Hamiltonian is performed after performing this reversible mapping $(\hat{\mathbf{w}},\hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w},\mathbf{p})$. The canonical transformation $\mathcal{X}(\mathbf{w},\mathbf{p})$ is given as [155, 99, 112]:

$$\begin{aligned}\hat{\mathbf{p}} &= \mathbf{p} - \frac{\epsilon^2}{12}\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} + \mathcal{O}(\epsilon^4)\\ \hat{\mathbf{w}} &= \mathbf{w} + \frac{\epsilon^2}{12}\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^4)\end{aligned} \tag{2.32}$$

where $(\hat{\mathbf{w}},\hat{\mathbf{p}})$ is found through fixed point[3] iterations as:

$$\begin{aligned}\hat{\mathbf{p}} &= \mathbf{p} - \frac{\epsilon}{24}\left[\mathrm{U}_{\mathbf{w}}(\mathbf{w}+\epsilon\mathbf{M}^{-1}\hat{\mathbf{p}}) - \mathrm{U}_{\mathbf{w}}(\mathbf{w}-\epsilon\mathbf{M}^{-1}\hat{\mathbf{p}})\right]\\ \hat{\mathbf{w}} &= \mathbf{w} + \frac{\epsilon^2}{24}\mathbf{M}^{-1}\left[\mathrm{U}_{\mathbf{w}}(\mathbf{w}+\epsilon\mathbf{M}^{-1}\hat{\mathbf{p}}) + \mathrm{U}_{\mathbf{w}}(\mathbf{w}-\epsilon\mathbf{M}^{-1}\hat{\mathbf{p}})\right]\end{aligned} \tag{2.33}$$

[3]Hessian approximated as: $\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} = \frac{1}{2\epsilon}\left[\mathrm{U}_{\mathbf{w}}(\mathbf{w}+\epsilon\mathbf{M}^{-1}\hat{\mathbf{p}}) - \mathrm{U}_{\mathbf{w}}(\mathbf{w}-\epsilon\mathbf{M}^{-1}\hat{\mathbf{p}})\right]$

After the leapfrog is performed, this mapping is reversed using post-processing via following fixed point iterations:

$$\begin{aligned}\mathbf{w} &= \hat{\mathbf{w}} - \frac{\epsilon^2}{24}\mathbf{M}^{-1}\big[\mathrm{U}_{\mathbf{w}}(\mathbf{w} + \epsilon\mathbf{M}^{-1}\hat{\mathbf{p}}) + \mathrm{U}_{\mathbf{w}}(\mathbf{w} - \epsilon\mathbf{M}^{-1}\hat{\mathbf{p}})\big] \\ \mathbf{p} &= \hat{\mathbf{p}} + \frac{\epsilon}{24}\big[\mathrm{U}_{\mathbf{w}}(\mathbf{w} + \epsilon\mathbf{M}^{-1}\hat{\mathbf{p}}) - \mathrm{U}_{\mathbf{w}}(\mathbf{w} - \epsilon\mathbf{M}^{-1}\hat{\mathbf{p}})\big]\end{aligned} \tag{2.34}$$

Theorem 2.7.2 guarantees that S2HMC, which uses the processed leapfrog integrator, satisfies detailed balance and thus leaves the target density invariant.

Theorem 2.7.2. *S2HMC satisfies detailed balance.*

Proof. To prove that S2HMC satisfies detailed balance, we need to show that the processing map $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \mathbf{p})$ commutes with reversal of momenta and preserves phase space volume. This will ensure that the resulting processed leapfrog integrator used in S2HMC is both sympletic and reversible. The detailed proof of this is provided in Sweet *et al.* [155], and we present this in Appendix C for ease of reference. □

Algorithm 5 Separable Shadow Hamiltonian Hybrid Monte Carlo Algorithm

Input: N, ϵ, L, w_{init}, $H(w,p)$, $\tilde{H}(w,p)$
Output: $(w)_{m=0}^{N}$, importance weights $= (b)_{m=0}^{N}$

$w_0 \leftarrow w_{\text{init}}$
for $m \to 1$ **to** N **do**
  $p_{m-1} \sim \mathcal{N}(0, \mathbf{M})$ ← **momentum refreshment**
  Apply the pre-processing mapping $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \mathbf{p})$ in equation (2.33)
  $p_m, w_m =$ **Leapfrog**$(p_{m-1}, w_{m-1}, \epsilon, L, \tilde{H})$ in equation (2.19)
  Apply the post-processing mapping $(\mathbf{w}, \mathbf{p}) = \mathcal{X}^{-1}(\hat{\mathbf{w}}, \hat{\mathbf{p}})$ in equation (2.34)
  $\delta H = \tilde{H}(w_{m-1}, p_{m-1}) - \tilde{H}(w_m, p_m)$
  $\alpha_m = \min(1, \exp(\delta H))$
  $u_m \sim \mathrm{Unif}(0,1)$
  $w_m =$ **Metropolis**$(\alpha, u_m, w_m, w_{m-1})$
  $b_m = \exp(-(H(w_m, p_m) - \tilde{H}(w_m, p_m)))$
end for

Once the samples are obtained from S2HMC as depicted in Algorithm 5, importance weights are calculated to allow for the use of the shadow canonical density rather than the true density. These weights are based on the differences between the true and shadow Hamiltonians as:

$$b_m = \exp[-(\mathrm{H}(\mathbf{w}, \mathbf{p}) - \hat{\mathrm{H}}(\mathbf{w}, \mathbf{p}))]. \tag{2.35}$$

Expectations of $f(\mathbf{w})$ are then computed as a weighted average. It is important to note that the weights in equation (2.35) should always be taken into account in every performance measure utilised to asses the performance of S2HMC. This is because non-uniform weights impact the overall utility of importance samplers.

Lines 4 to 6 in Algorithm 5 represent the processed leapfrog integrator. This scheme degenerates to the traditional leapfrog integration scheme when the shadow Hamiltonian is equal to the true Hamiltonian.

2.8 No-U-Turn Sampler Algorithm

We have yet to address how one selects the step size ϵ and trajectory length L parameters of samplers based on Hamiltonian dynamics such as HMC, MHMC and S2HMC. These parameters significantly influence the sampling performance of the algorithms [73]. A significant step size typically results in most of the produced samples being rejected, while a small step size results in slow convergence and mixing of the chain [73]. When the trajectory length is too small, then the method displays random walk behaviour, and when the trajectory length is considerable, the algorithm wastes computational resources [73]. The NUTS algorithm of Hoffman and Gelman [73] automates the tuning of the HMC step size and trajectory length parameters.

In NUTS, the step size parameter is tuned through primal-dual averaging during an initial burn-in phase. A user specified MH acceptance rate δ is targeted via primal-dual averaging given as [97, 8]:

$$\begin{aligned} \epsilon_{t+1} &\leftarrow \mu - \frac{\sqrt{t}}{\gamma} \frac{1}{t + t_0} \sum_{i=1}^{t} H_i \\ \bar{\epsilon}_{t+1} &\leftarrow \eta_t \epsilon_{t+1} + (1 - \eta_t)\bar{\epsilon}_t \end{aligned} \tag{2.36}$$

where μ is a free parameter that ϵ_t gravitates to, γ controls the convergence towards μ, η_t is a decaying rate of adaptation in line with [8], and H_t is the difference between the target acceptance rate and the actual acceptance rate [99, 73]. The primal-dual averaging updates are such that [99]:

$$\mathbb{E}[H_t] = \mathbb{E}[\delta - \alpha_t] = 0. \tag{2.37}$$

This has the effect of updating the step size towards the target acceptance rate δ. Hoffman and Gelman [73] found that setting $\mu = \log(10\epsilon_0)$, $\bar{\epsilon}_0 = 1$, $\bar{H}_0 = 0$, $\gamma = 0.05$, $t_0 = 10$, $\kappa = 0.75$ with ϵ_0 being the initial step size results in good performance across various target posterior distributions. These are the settings that we utilise in this thesis with $\epsilon_0 = 0.0001$.

In the NUTS methodology, the trajectory length parameter is automatically tuned by iteratively doubling the trajectory length until the Hamiltonian becomes infinite or the chain starts to trace back [98, 2]. That is, when the last proposed position state $\mathbf{w}^*$ starts becoming closer to the initial position $\mathbf{w}$. This happens if: $(\mathbf{w}^* - \mathbf{w}) \times \mathbf{p} \leq 0$. This approach, however, violates the detailed balance condition. To overcome the violation of detailed balance in NUTS, a path of states $\mathcal{B}$ is generated such that its size is determined by the termination criterion [2, 73]. The set $\mathcal{B}$ has the following property, which is required for detailed balance [2, 73]:

$$P(\mathcal{B}|\mathbf{z}) = P(\mathcal{B}|\mathbf{z}') \quad \forall \mathbf{z}, \mathbf{z}' \in \mathcal{B}. \tag{2.38}$$

where $\mathbf{z} = (\mathbf{w}, \mathbf{p})$ is the current state. That is, the probability of generating $\mathcal{B}$, starting from any of its members is the same [2, 73]. Having $\mathcal{B}$, the current state $\mathbf{z}$ and a slice variable u that is uniformly drawn from the interval $[0, \pi_{\mathbf{Z}}(\mathbf{z})]$, a set of chosen states $\mathcal{C}$ [2, 73]:

$$\mathcal{C} := \{\mathbf{z}' \in \mathcal{B} \quad s.t \quad \pi_{\mathbf{Z}}(\mathbf{z}) \geq u\} \tag{2.39}$$

is constructed from which the next state is drawn uniformly.

Algorithm 6 shows the pseudo-code for NUTS with dual averaging, showing how the trajectory length L and the step size ϵ are set automatically. The NUTS algorithm has a main *while* loop that contains the termination criterion and is presented in Algorithm 6, as well as an algorithm for iteratively doubling the trajectory length which is shown

in Algorithm 7. The overall methodology of generating a single sample from NUTS can be summarised as follows, with the reader being referred to Hoffman and Gelman [73] for more information on the NUTS algorithm:

1. Set the initial states - which is the **Input** step in Algorithm 6
2. Generate the auxiliary momentum and slice variables as shown in line 2 of Algorithm 6
3. Generate the set of chosen states $\mathcal{C}$ via the doubling procedure in Algorithm 7. Note the use of the leapfrog integration scheme in line 2 in Algorithm 7
4. Each proposal is accepted or rejected in line 12 of Algorithm 6. Note that this is not the typical Metropolis acceptance probability as the proposal is chosen from multiple candidates in $\mathcal{C}$
5. Update the step size via dual averaging in lines 17 - 21 in Algorithm 6. Note that this step size is no longer updated post the burn-in period which is indicated by M^{adapt}.

One then repeats steps 2 through 5 to generate the required number of samples N.

Algorithm 6 No-U-Turn Sampler With Primal-Dual Averaging Algorithm

Input: M, M^{adapt}, ϵ_0, w^0, $\mu = \log(10\epsilon_0)$, $\bar{\epsilon}_0 = 1$, $\bar{H}_0 = 0$, $\gamma = 0.05$, $t_0 = 10$, $\kappa = 0.75$ and $\epsilon_0 = 0.0001$

Output: $(w)_{m=0}^{N}$

for $m \to 1$ **to** M **do**
  $p^{m-1} \sim \mathcal{N}(0, \mathbf{I})$, $u \sim \text{Unif}[0, \exp(-H(w^{m-1}, p^{m-1}))]$
  $w^- = w^{m-1}$, $w^+ = w^{m-1}$, $w^- = w^{m-1}$, $p^- = p^{m-1}$, $p^+ = p^{m-1}$, $j = 0$, $n = 1$, $s = 1$.
  while s =1 **do**
    $v \sim \text{Unif}[-1, 1]$
    if $v = -1$ **then**
      $w^-, p^-, -, -, w', p', n', s', \alpha, n_\alpha =$ **BuildTree**$(w^-, p^-, u, v, j, \epsilon_{m-1}, w^{m-1}, p^{m-1})$ in Algorithm 7
    else
      $-, -, w^+, p^+, w', p', n', s', \alpha, n_\alpha =$ **BuildTree**$(w^+, p^+, u, v, j, \epsilon_{m-1}, w^{m-1}, p^{m-1})$ in Algorithm 7
    end if
    if $s' = 1$ **then**
      With prob. $\min(1, \frac{n'}{n})$, set $w^m = w'$, $p^m = p'$
    end if
    $n = n + n'$, $j = j + 1$
    $s = s'\mathbf{I}[(w^+ - w^-)p^- \geq 0]\mathbf{I}[(w^+ - w^-)p^+ \geq 0]$
  end while
  if $m < M^{adapt}$ **then**
    $\bar{H}_m = \left(1 - \frac{1}{m+t_0}\right)\bar{H}_{m-1} + \frac{1}{m+t_0}(\delta - \alpha_m)$
    $\log \epsilon_m = \mu - \frac{\sqrt{m}}{\gamma}\bar{H}_m$, $\log \bar{\epsilon}_m = m^{-\kappa}\log \epsilon_m + (1 - m^{-\kappa})\log \bar{\epsilon}_{m-1}$
  else
    $\epsilon_m = \bar{\epsilon}_{M^{adapt}}$
  end if
end for

Algorithm 7 Algorithm For Iteratively Doubling The Trajectory Length

function BuildTree$(w, p, u, v, j, \epsilon, w^0, p^0)$

Output: $w^-, p^-, w^+, p^+, w', p', n', s', \alpha', n'_\alpha$

if $j = 0$ **then**
  $w', p' =$ **Leapfrog**$(p,\ w,\ \epsilon v)$
  $n' = \mathbf{I}[u < \exp(-H(w', p')],\ s' = \mathbf{I}[u < \exp(\Delta_{max} - H(w', p')]$
  $n_\alpha = 1,\ \alpha = \min(1, \exp(H(w^0, p^0) - H(w', p')))$
  return $w', p', w', p', w', p', n', s', \alpha, n_\alpha$
else
  $w^-,^-, w^+,^+, w', p', n', s', \alpha', n'_\alpha =$ BuildTree$(w, p, u, v, j-1, \epsilon, w^0, p^0)$
  if s'=1 **then**
    if $v = -1$ **then**
      $w^-, p^-, -, -, w'', p'', n'', s'', \alpha'', n''_\alpha =$ BuildTree$(w^-, p^-, u, v, j-1, \epsilon, w^0, p^0)$
    else
      $-, -, w^+, p^+, w'', p'', n'', s'', \alpha'', n''_\alpha =$ BuildTree$(w^+, p^+, u, v, j-1, \epsilon, w^0, p^0)$
    end if
    With prob. $\frac{n'}{n'+n''}$, set $w' = w'',\ p' = p''$
    $\alpha' = \alpha' + \alpha'',\ n' = n' + n'',\ n'_\alpha = n'_\alpha + n''_\alpha$
    $s' = s''\mathbf{I}[(w^+ - w^-)p^- \geq 0]\mathbf{I}[(w^+ - w^-)p^+ \geq 0]$
  end if
  return $w^-, p^-, w^+, p^+, w', p', n', s', \alpha', n'_\alpha$
end if

2.9 Antithetic Hamiltonian Monte Carlo

The coupling of MCMC chains has been used as a theoretical tool to prove convergence behaviour of Markov chains [146, 80, 79, 78, 20]. Johnson [80] presents a convergence diagnostic for MCMC methods that is based on coupling a Markov chain with another chain that is periodically restarted from fixed parameter values. The diagnostic provides an informal lower bound on the ESS rates of the MCMC chain. Jacob *et al.* [78] remove the bias in MCMC chains by using couplings of Markov chains using a telescopic sum

argument. The resulting unbiased estimators can then be computed independently in parallel. More recently, MCMC couplings have been studied for HMC with Heng and Jacob [69] proposing an approach that constructs a pair of HMC chains that are coupled in such a way that they meet after some random number of iterations. These chains can then be combined to create unbiased chains. Unlike the approach of Heng and Jacob [69] and other authors [136, 78], Bou-Rabee *et al.* [20] present a new coupling algorithm where the momentum variable is not shared between the coupled chains.

In this thesis, we explore methods that use the results from coupling theory to create anti-correlated HMC based chains where the momentum variable is shared between the chains. We create these anti-correlated chains by running the second chain with the auxiliary momentum variable having the opposite sign of the momentum of the first chain. These chains also share the random uniform variable in the MH acceptance step. When these two chains anti-couple strongly, taking the average of the two chains results in estimators that have lower variance, or equivalently, these chains produce samples that have higher ESSs than their non-antithetic counterparts.

The work that is closely related to ours is the recent article by Piponi *et al.* [136], where the authors introduce the antithetic HMC variance reduction technique. The antithetic HMC samplers have an advantage over antithetic Gibbs samplers. Antithetic HMC methods apply to problems where conditional distributions are intractable and where Gibbs sampling may mix slowly [136]. The pseudo-code for the antithetic HMC method of Piponi *et al.* is shown in Algorithm 8. The difference between the antithetic HMC algorithm and the original HMC method is that the momentum variable (see line 3 and 4 in Algorithm 8) and the uniform random variable in the MH acceptance step (see line 13 and 14 in Algorithm 8) are shared between the two chains.

Suppose we have two random variables X and Y, that are not necessarily independent, with the same marginal distribution U, we have that:

$$\text{Var}\left[\frac{f(X)+f(Y)}{2}\right] = \frac{1}{4}\left[\text{Var}f(X) + \text{Var}f(Y)\right] + \frac{1}{2} \times \text{Cov}\left[f(X), f(Y)\right]. \tag{2.40}$$

When $f(X)$ and $f(Y)$ are negatively correlated, the average of the two random variables will produce a lower variance than the average of two independent variables. This equation forms the basis for the antithetic sampling algorithms that we introduce in this

thesis.

Algorithm 8 Antithetic Hamiltonian Monte Carlo Algorithm

Input: N, ϵ, L, w^x_{init}, w^y_{init}, $H(w, p)$

Output: $(w^x)_{m=0}^N$, $(w^y)_{m=0}^N$

1: $w^x_0 \leftarrow w^x_{\text{init}}$

2: $w^y_0 \leftarrow w^y_{\text{init}}$

3: **for** $m \rightarrow 1$ **to** N **do**

4: $\quad p^x_{m-1} \sim \mathcal{N}(0, \mathbf{M})$

5: $\quad p^y_{m-1} = -p^x_{m-1} \leftarrow$ **momentum shared between the two chains**

6: $\quad p^x_m, w^x_m =$ **Leapfrog**$(p^x_{m-1}, w^x_{m-1}, \epsilon, L, H)$

7: $\quad p^y_m, w^y_m =$ **Leapfrog**$(p^y_{m-1}, w^y_{m-1}, \epsilon, L, H)$

8: $\quad \delta H^x = H(w^x_{m-1}, p^x_{m-1}) - H(w^x_m, p^x_m)$

9: $\quad \delta H^y = H(w^y_{m-1}, p^y_{m-1}) - H(w^y_m, p^y_m)$

10: $\quad \alpha^x_m = \min\left(1, \exp\left(\delta H^x\right)\right)$

11: $\quad \alpha^y_m = \min\left(1, \exp\left(\delta H^y\right)\right)$

12: $\quad u_m \sim \text{Unif}(0, 1) \leftarrow$ **uniform random variable shared in line 13 and 14 below**

13: $\quad w^x_m =$ **Metropolis**$(\alpha^x_m, u_m, w^x_m, w^x_{m-1})$

14: $\quad w^y_m =$ **Metropolis**$(\alpha^y_m, u_m, w^y_m, w^y_{m-1})$

15: **end for**

The full benefit of antithetic sampling is most significant when the target distribution $\mathbf{U}(\mathbf{w})$ is symmetrical about some vector [47, 136]. For the majority of target distribution of interest to machine learning researchers, the symmetry holds only approximately [136]. The approximate symmetry results in approximate anti-coupling; that is, we will not observe perfect anti-correlation in practice as would be the case if $\mathbf{U}(\mathbf{w})$ was symmetric. However, the approximate anti-coupling nonetheless still provides decent variance reduction in practice [136]. We rely on these results to propose new antithetic versions

of MHMC and S2HMC later in this thesis.

2.10 Conclusion

In this chapter, we presented a review of the MCMC methods that form the core of the algorithms that we propose in this thesis. In the following chapters, we will introduce new MCMC methods and apply the Bayesian inference framework in domains that are socially relevant within the South African context.

Chapter 3

Sampling Benchmarks, Application Areas and Performance Metrics

3.1 Introduction

This chapter outlines the benchmark problems, datasets, and performance metrics that we use to assess the performance of the novel MCMC algorithms that we propose in this thesis. We first outline the benchmark target posterior distributions being the Banana shaped distribution, multivariate Gaussian distributions of varying dimensionality, and Neal's [123] funnel density. We then consider real-world financial market datasets modelled using Merton [100] Jump-Diffusion Process (JDP)s, and benchmark datasets modelled using Bayesian Logistic Regression (BLR) and BNNs. Furthermore, we introduce into the literature a new real-world and public interest dataset on the South African municipal financial statement audit outcomes. This dataset provides significant insight into the financial stability of South African local government entities. We perform an exploratory data analysis using a SOM to identify the key financial ratios when modeling audit opinions. Our analysis using the SOM shows that the *current ratio*, *debt to total operating revenue* and the *net surplus profit margin* are important financial ratios when modelling audit outcomes of South African municipalities. The material in this chapter has been published in the following works:

- **Mongwe, W.T.** and Malan, K.M., 2020. *A survey of automated financial state-*

ment fraud detection with relevance to the South African context. South African Computer Journal, vol. 32, no. 1, pp. 74-112.

- **Mongwe, W.T.** and Malan, K.M., 2020. *The Efficacy of Financial Ratios for Fraud Detection Using Self Organising Maps.* In 2020 IEEE Symposium Series on Computational Intelligence (SSCI).pp. 1100-1106. IEEE.

3.2 Benchmark Problems and Datasets

In this section, we present the benchmark posterior distributions that we consider in this thesis. These benchmark target posterior densities have been extensively used in the literature [53, 140, 156, 116, 68, 113, 112], including in the seminal work of Girolami and Caldehead [53]. We also present the benchmark datasets that we model using JDPs, BLR and BNNs respectively.

3.2.1 Banana shaped distribution

The Banana shaped density of Haario *et al.* [62] is a 2-dimensional non-linear target and provides an illustration of a distribution with significant ridge-like structural features that commonly occur in non-identifiable models [62, 140, 68]. The likelihood and prior distributions are as follows:

$$y|\mathbf{w} \sim \mathcal{N}(w_1 + w_2^2 = 1, \sigma_y^2), \quad w_1, w_2 \sim \mathcal{N}(0, \sigma_{\mathbf{w}}^2) \tag{3.1}$$

We generated one hundred data points for y with $\sigma_y^2 = 4$ and $\sigma_{\mathbf{w}}^2 = 1$. Due to independence of the data and parameters, the posterior distribution is proportional to:

$$\prod_{i=1}^{i=N} p(y_k|\mathbf{w})p(w_1)p(w_2). \tag{3.2}$$

where $N = 100$ is the number of observations.

3.2.2 Multivariate Gaussian distributions

The task is to sample from D-dimensional Gaussian distributions $\mathcal{N}(0, \mathbf{\Sigma})$ with mean zero and covariance matrix $\mathbf{\Sigma}$. The covariance matrix $\mathbf{\Sigma}$ is diagonal, with the standard

deviations simulated from a log-normal distribution with mean zero and unit standard deviation. For the simulation study, we consider the number of dimensions D to be in the set $\{10, 50\}$.

3.2.3 Neal's funnel density

Neal [123] introduced the funnel distribution as an example of a density that illustrated the issues that arise in Bayesian hierarchical and hidden variable models [18, 68]. The variance of the parameters is treated as a latent variable with a log-normal distribution [18, 68]. In this thesis, we consider the number of dimensions $D = 25$. The target density of interest is:

$$P(v, \mathbf{x}) = \mathcal{N}(v|\mu, \sigma^2) \prod_{i=1}^{D} \mathcal{N}(x_i|0, \exp(v)). \tag{3.3}$$

where $\mu = 0, \sigma = 3$, $v \sim \mathcal{N}(\mu, \sigma^2)$ and $\mathbf{x_i} \in \mathbb{R}$. Note that v and $\mathbf{x}$ are the parameters of interest.

3.2.4 Merton jump-diffusion process model

It is well documented that financial asset returns do not have a Gaussian distribution, but instead have fat tails [36, 108]. Stochastic volatility models, Levy models, and combinations of these models have been utilised to capture the leptokurtic nature of asset return distributions [3, 6, 101, 158]. JDPs were introduced into the financial markets literature by Merton and Press in the 1970s [108, 101, 138]. In this thesis, we study the JDP model of Merton [101], which is a one-dimensional Markov process $\{X_t, t \geq 0\}$ with the following dynamics as described in Mongwe [108]:

$$d \ln X_t = \left(a - \frac{1}{2}b^2\right) dt + b dZ_t + d\left(\sum_{i=1}^{K_t} J_i\right) \tag{3.4}$$

where a and b are drift and diffusion terms, Z_t and K_t are the Brownian motion process and Poisson process with intensity λ respectively, and $J_i \sim \mathcal{N}(a_{jump}, b^2_{jump})$ is the size of the ith jump. As shown in Mongwe [108], the density implied by the dynamics in

equation (3.4) is:

$$\mathbb{P}(\ln X(t+\tau) = x_2 | \ln X(t) = x_1) = \sum_{n=0}^{\infty} \frac{e^{-\lambda\tau}(\lambda\tau)^n}{n!} \frac{\phi\left(\frac{x_2 - x_1 - (a\tau + n a_{jump})}{\sqrt{b^2\tau + n b_{jump}^2}}\right)}{\sqrt{b^2\tau + n b_{jump}^2}} \tag{3.5}$$

with x_2 and x_1 being realisations of $\ln X(t)$ at times $t+\tau$ and t, and ϕ is a Gaussian probability density function. The resultant likelihood is multi-modal as it is an infinite combination of Gaussian random variables, where the Poisson distribution produces the mixing weights [108]. In this thesis, we truncate the infinite summation in equation (3.5) to the first 10 terms as done in [108]. Furthermore, the JDP model is calibrated to historical financial market returns data as outlined in Table 3.2.

3.2.5 Bayesian logistic regression

We utilise BLR to model the real world binary classification datasets in Table 3.2. The negative log-likelihood $l(\mathrm{D}|w)$ function for logistic regression is given as:

$$l(\mathrm{D}|\mathbf{w}) = \sum_{i}^{N} y_i \log(\mathbf{w}^T x_i) + (1 - y_i)\log(1 - \mathbf{w}^T x_i) \tag{3.6}$$

where D is the data and N is the number of observations. The log of the unnormalised target posterior distribution is given as:

$$\ln p(\mathbf{w}|\mathrm{D}) = l(\mathrm{D}|\mathbf{w}) + \ln p(\mathbf{w}|\alpha) \tag{3.7}$$

where $\ln p(\mathbf{w}|\alpha)$ is the log of the prior distribution on the parameters given the hyper-parameters α. The parameters $\mathbf{w}$ are modelled as having Gaussian prior distributions with zero mean and standard deviation $\alpha = 10$ as in Girolami and Caldehead [53].

3.2.6 Bayesian neural networks

Artificial neural networks are learning machines that have been extensively employed as universal approximators of complex systems with great success [99, 112]. This thesis focuses on MLPs with one hidden layer, five hidden units, and a single output neuron. An example of this architecture is shown in Figure 3.1. The MLPs are used to model

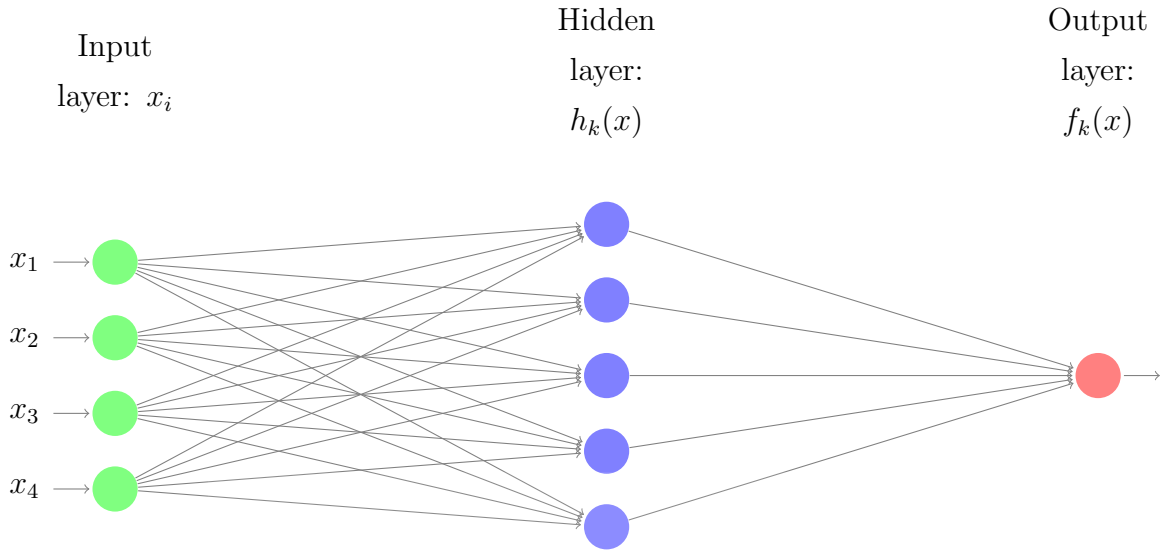


Figure 3.1: An illustration of the data flow in a MLP. In this thesis, we limit our investigations to MLPs with one hidden layer, five hidden units and a single output.

the real-world benchmark datasets outlined in Table 3.2. These datasets are regression datasets, with the negative log-likelihood being the sum of squared errors.

The outputs of a network with a single output as depicted in Figure 3.1 are defined as:

$$\begin{aligned} f_k(x) &= b_k + \sum_j v_{jk} h_j(x) \\ h_j(x) &= \Psi\left(a_j + \sum_i w_{ij} x_i\right) \end{aligned} \tag{3.8}$$

where w_{ij} is the weight connection for the i^{th} input to the j^{th} hidden neuron and v_{jk} is the weight connection between the j^{th} hidden neuron to the k^{th} output neuron [99]. In this thesis we only consider single output MLPs so that $k = 1$. Note that Ψ is the activation function which produces the required non-linearity [99], and we set Ψ to be the hyperbolic tangent activation function in this thesis.

3.2.7 Benchmark datasets

We present the details of the financial and real-world benchmark datasets used in this thesis in Table 3.2. The datasets are subdivided based on the type of model utilised to model the datasets. The datasets consist of historical financial market data that we use to calibrate the jump-diffusion process model, as well as real-world benchmark classification datasets, which we model using BLR, and real-world benchmark regression datasets modeled using BNNs.

Jump-diffusion process datasets: We calibrate the JDP to three datasets. These are real-world datasets across different financial markets. The real-world financial market datasets consist of daily prices, which we convert into log-returns, that were obtained from Google Finance[1] [56]. The specifics of the datasets are as follows:

- *Bitcoin dataset*: 1 461 daily data points for the cryptocurrency (in USD) from 1 Jan 2017 to 31 Dec 2020.
- *S&P 500 dataset*:1 007 daily data points for the stock index from 1 Jan 2017 to 31 Dec 2020.
- *USDZAR dataset*: 1 425 daily data points for the currency from 1 Jan 2017 to 31 Dec 2020.

Note that the formula used to calculate the log-returns is given as:

$$r_i = \log\left(S_i/S_{i-1}\right) \tag{3.9}$$

where r_i is the log-return on day i and S_i is the stock or currency level on day i. The descriptive statistics of the dataset are shown in Table 3.1. This table shows that the USDZAR dataset has a very low kurtosis, suggesting that it has very few (if any) jumps when compared to the other two datasets.

Bayesian logistic regression datasets: There are four datasets that we modeled using BLR. All the datasets have two classes and thus present a binary classification problem. These datasets are available on the UCI machine learning repository[2]. The specifics of the datasets are:

[1] https://www.google.com/finance/

[2] https://archive.ics.uci.edu/ml/index.php

Table 3.1: Descriptive statistics for the financial markets datasets.

Dataset	mean	standard deviation	skew	kurtosis
S&P 500 Index	0.00043	0.01317	-1.159	21.839
Bitcoin	0.00187	0.04447	-0.925	13.586
USDZAR	0.00019	0.00854	0.117	1.673

- *Pima Indian Diabetes dataset* - This dataset has seven features and a total of 532 observations. The aim of this dataset is to predict if a patient has diabetes based on diagnostic measurements made on the patient [102].
- *Heart dataset* - This dataset has 13 features and 270 data points. The purpose of the dataset is to predict the presence of heart disease based on medical tests performed on a patient [102].
- *Australian credit dataset* - This dataset has 14 features and 690 data points. The objective for this dataset is to assess applications for credit cards [102].
- *German credit dataset* - This dataset has 25 features and 1 000 data points. The aim for this datatset was to classify a customer as either good or bad credit [102].

Bayesian Neural Network Datasets: There are four datasets that we model using BNNs. All the datasets are regression datasets and as with the BLR datasets, are also available on the UCI machine learning repository. The specifics of the datasets is as follows:

- *Power dataset* - This data set has 9 568 observations and 4 features.
- *Airfoil dataset* - This dataset has 5 features and 1 503 observations.
- *Concrete dataset* - This dataset has 1 030 observations and 8 features [169].
- *Power dataset* - This dataset has 9 features and 45 730 observations.

Table 3.2: Real-world datasets used in this thesis. N represents the number of observations. BJDP is Bayesian JDP, BLR is Bayesian Logistic Regression, and BNN represents Bayesian Neural Networks. D represents the number of model parameters.

Dataset	Features	N	Model	D
S&P 500	1	1 007	BJDP	5
Bitcoin	1	1 461	BJDP	5
USDZAR	1	1 425	BJDP	5
Pima	7	532	BLR	8
Heart	13	270	BLR	14
Australian	14	690	BLR	15
German	24	1 000	BLR	25
Power	4	9 568	BNN	31
Airfoil	5	1 503	BNN	36
Concrete	8	1 030	BNN	51
Protein	9	45 730	BNN	56

3.2.8 Processing of the datasets

The JDP datasets used a time series of log returns as the input. The features for the BLR and BNN datasets were normalised. A random 90-10 train-test split was used for all the datasets. The train-test split is based on a cutoff date that confines 90% of the data into the training set for the time series datasets.

3.3 Municipal Financial Statement Audit Outcome Dataset

This section introduces the municipal financial statement audit outcome dataset, which we use to predict the financial statement audit outcomes for South African local government entities. We also present the results of the exploratory data analysis that we performed using unsupervised SOM on this dataset. SOM are used due to their visual nature and the resulting accessibility of information to decision-makers. This exercise

revealed which features are essential when modelling the dataset, giving great insight into the dataset.

3.3.1 Overview of financial statement fraud

A 2018 study by audit firm PricewaterhouseCoopers [139] states that 77% of South African companies surveyed have experienced some form of economic crime. According to the report, South Africa had the highest percentage of economic crime in the world in 2018, with Kenya and France coming in at positions two and three, respectively. One of the companies' economic crimes reported to have been experienced is accounting fraud, a subset of which is management fraud or Financial Statement Fraud (FSF). In South Africa, accounting fraud experienced by the companies surveyed increased from 20% in 2016 to 22% in 2018 [139]. This suggests that accounting fraud is becoming more common in South Africa and poses a risk to the stability of South African capital markets.

Accounting fraud cases prominent over the last two decades include (1) Enron, which was an American natural gas company that used creative accounting to make it appear as if the firm was growing, but eventually lost over $60 billion in market capitalisation from January 2001 to January 2002 when the allegations of fraud emerged [67, 94], (2) Steinhoff, which is a South African retailer that lost over R200 billion in market capitalisation in the space of two weeks after it emerged that accounting fraud had allegedly been perpetrated by the management of the firm [38]. Furthermore, the Public Investment Corporation, which manages the pension fund assets for South African government employees, lost at least R19 billion due to its direct and indirect investments in Steinhoff [40, 137].

The financial statements consist of reports such as the income statement, balance sheet, and cash flow statements. These statements are usually summarised into financial ratios and used by different stakeholders for various purposes. For example, the government would use financial statements to determine the tax payable to the state by the entity.

FSF occurs when the financial statements of an entity are manipulated to make the entity appear to be in a better financial state than is the case [150]. The management

of the entity often perpetrates this manipulation, and at times with the support and knowledge of the auditors of the entity [23, 67]. Examples of fraud that can be present in a financial statement of an entity include the manipulation of the entity's earnings and the omission of material information [150, 134].

When using machine learning algorithms to analyse and detect FSF, a critical element is a definition of what constitutes fraud. The most common definitions of fraud, as shown in Appendix A, used in the literature are: 1) results of investigations by authorities such as the Securities and Exchange Commission in the USA [13, 152], and 2) by qualified audit opinion [106, 109]. In this thesis, the audit opinion expressed by the Auditor General of South Africa (AG-SA) is used as the definition of FSF.

The audit opinion expressed by the AG-SA on the financial statements of South African municipalities falls broadly into the following categories [10]:

- *Clean or unqualified audit opinion*—The financial statements contain no material misstatements. Note that this does not necessarily mean there was no fraud.
- *Qualified audit opinion*—The financial statements contain material misstatements in specific amounts, or there is insufficient evidence to conclude that the amounts are not materially misstated.
- *Adverse audit opinion*—The financial statements contain material misstatements. This, however, does not necessarily mean that there was fraud present.
- *Disclaimer audit opinion*—The municipality provided insufficient evidence in the form of documentation on which to base an audit opinion.

For this thesis, we consider the statements of a municipality to be a fraudulent instance if the audit opinion is not a clean or unqualified audit, which is consistent with other studies in the literature [106, 39] as highlighted in Appendix A. Thus, we consider a financial statement fraudulent if the AG-SA expressed a qualified, adverse, or disclaimer audit opinion, and a financial statement is considered not fraudulent when it receives a clean or an unqualified audit opinion. In addition, we limit our analysis to using financial ratios. We do not use other features such as the text in the financial statements or the credit ratings of the entity. We plan to incorporate such information in future work.

3.3.2 Self organising maps

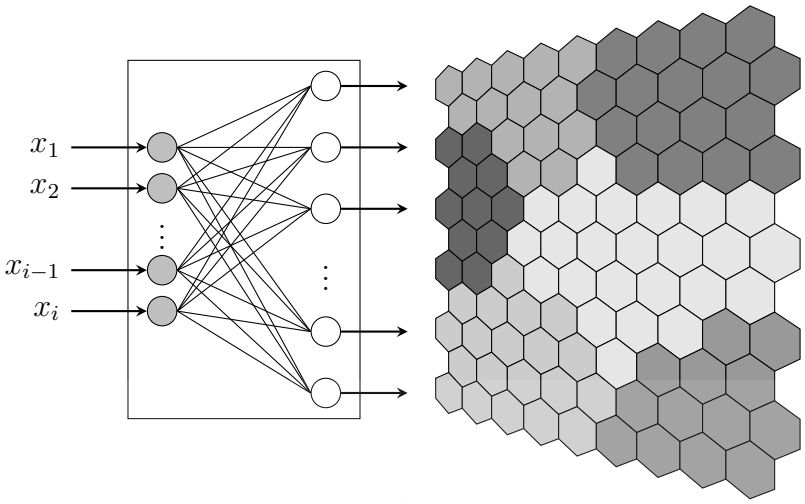


Figure 3.2: An illustration of the data flow in a Kohonen [85] SOM.

We now present our chosen unsupervised tool of analysing the dataset being the SOM. The SOM was invented by Teuvo Kohonen in 1981/2, and various versions and generalisations of the SOM have been developed since then [85]. The SOM is a two-layer neural network that can learn to represent distributions of the data presented to it. [85, 132, 107]. The two layers, being the input and output layers, are fully connected in that each neuron in the input layer is connected to every neuron in the output layer [107, 85, 132]. These connections from the input to output layer store an exemplar vector [132]. A SOM is equipped with a structure, typically a hypercube or hexagonal lattice as shown in Figure 3.2.

SOMs are preferable to other clustering approaches due to their ability to learn online, i.e., learning additional cases when they arise, and their ability to use a topological map for visualisation - which assists in the identification and discovery of clusters [132]. This makes SOMs easy to interpret for interested stakeholders.

The output layer of the SOM can be viewed as a 2D grid as illustrated in Figure 3.2. The SOM preserves the topological structure of the data so that inputs that are close in feature space will be close on the 2D grid [85, 132, 107]. The SOM is unsupervised as the training process does not require labels.

The SOM implementation used in this section was the **R** Kohonen package [165, 166]. The parameters used for the SOM was a map with a 20 by 20 grid, and the SOM was trained for 20 000 iterations. The learning rate was set to 0.05. After training the SOM, k-means clustering with 10 clusters was performed. The 10 clusters were chosen because they gave the best visual representation when compared to setting the clusters to be 2 through 20 for the dataset.

3.3.3 Data description

The data used is obtained from South African municipalities' audited financial statement data for the period 2010 to 2018. The years 2010 to 2012 had missing data and were excluded from our analysis. The data was sourced from the South African National Treasury website[3] [120]. The dataset had a total of 1 560 records from 2012 to 2018. This dataset contains, for each municipality in South Africa, the income statement, balance sheet, and cash flow statement for each year. The dataset also contains reports that outline audit opinions, conditional grants, and capital expenditure. In this thesis, we create financial ratios based on each municipality's income statement, balance sheet statement, cash flow statement, and capital acquisition report. The conversion of the financial statements into financial ratios is discussed in Section 3.3.4.

The municipalities in South Africa are governed by the Municipal Finance Management Act, and are audited by the AG-SA [11]. The distribution of audit opinions for the dataset shows that the unqualified data instances are just under 55%, with the rest being qualified or worse cases. This distribution also means that there is no large class-imbalance problem for this dataset. Note that outstanding audit opinions, which were only 3 in the data set, are assigned to the disclaimer audit opinion class [10].

3.3.4 Financial ratio calculation

The financial ratios used in this thesis are based on the municipal circular number 71 issued by the National Treasury of the Republic of South Africa [119]. The financial ratios considered can be grouped into two broad categories, namely financial performance

[3] https://municipaldata.treasury.gov.za/

and financial position. The financial performance ratios can be further broken down into efficiency, distribution losses, revenue management, expenditure management, and grant dependency. The financial position ratios can be subdivided into asset management/utilisation, debtors management, liquidity management, and liability management. The specific ratios considered in this thesis are [119]:

1. *Debt to Community Wealth/Equity* - Ratio of debt to the community equity. The ratio is used to evaluate a municipality's financial leverage.

2. *Capital Expenditure to Total Expenditure* - Ratio of capital expenditure to total expenditure.

3. *Impairment of PPE, IP and IA* - Impairment of Property, Plant and Equipment (PPE) and Investment Property (IP) and Intangible Assets (IA).

4. *Repairs and Maintenance as a percentage of PPE +IP* - The ratio measures the level of repairs and maintenance relative to assets.

5. *Debt to Total Operating Revenue* - The ratio indicates the level of total borrowings in relation to total operating revenue.

6. *Current Ratio* - The ratio is used to assess the municipality's ability to pay back short-term commitments with short-term assets.

7. *Capital Cost to Total Operating Expenditure* - The ratio indicates the cost of servicing debt relative to overall expenditure.

8. *Net Operating Surplus Margin* - The ratio assesses the extent to which the entity generates operating surpluses.

9. *Remuneration to Total Operating Expenditure* - The ratio measures the extent of remuneration of the entity's staff to total operating expenditure.

10. *Contracted Services to Total Operating Expenditure* - This ratio measures how much of total expenditure is spent on contracted services.

11. *Own Source Revenue to Total Operating Revenue* - The ratio measures the extent to which the municipality's total capital expenditure is funded through internally generated funds and borrowings.

12. *Net Surplus / Deficit Water* - This ratio measures the extent to which the municipality generates surplus or deficit in rendering water service

13. *Net Surplus / Deficit Electricity* - This ratio measures the extent to which the municipality generates surplus or deficit in rendering electricity service.

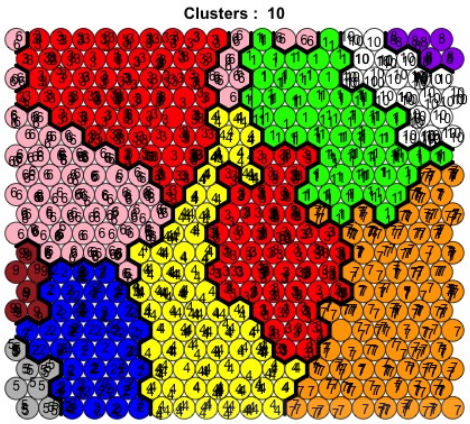
The first six ratios above are measures of financial position, while the rest are measures of financial performance. Table 3.3 provides descriptive statistics of the financial ratios in the dataset. None of the financial ratios had missing values. Furthermore, the financial ratios were not strongly correlated, with the maximum absolute pairwise linear correlations being less than 0.5.

Feature selection was performed on the financial ratios. The three ratios that had the largest correlation with the output variable, that is audit opinions, were selected. The selected ratios were *current ratio* (ratio 6), *debt to total operating revenue* (ratio 5) and *net operating surplus margin* (ratio 8). Note that the number of ratios to use, which is three, was based on the number that produced the most visually explainable results from the SOM, while simultaneously avoiding SOM clusters that had single data instances. The features were normalised before being passed into the SOM for training.

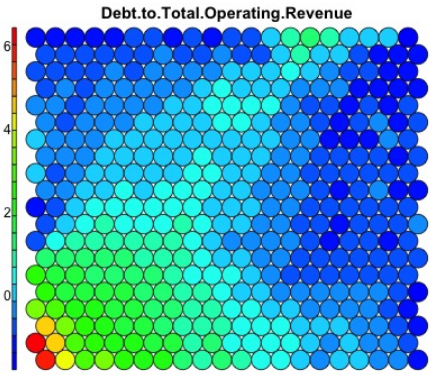
3.3.5 Exploratory data analysis

The trained SOM, after applying k-means clustering with 10 clusters, is displayed in Figure 3.3(a). The purpose of the clustering is to identify groups of instances, in this case municipal financial statements, with a similar financial ratios profile. Table 3.4 presents the results of labelling each of the data instances in each of the clusters by their associated audit opinions. Table 3.5 shows the percentage splits of unqualified (i.e., non-fraudulent) and not-unqualified (qualified, disclaimer, and adverse opinions - i.e., fraudulent) in each of the ten classes, sorted by the percentage of unqualified instances.

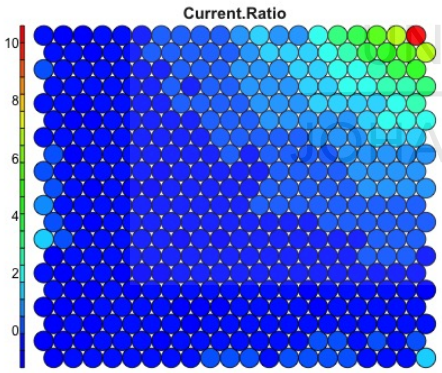
Table 3.5 shows that clusters 1, 8, and 10 (top three clusters in the table) consist of predominantly unqualified / non-fraudulent instances, while clusters 2, 5, and 9 (bottom



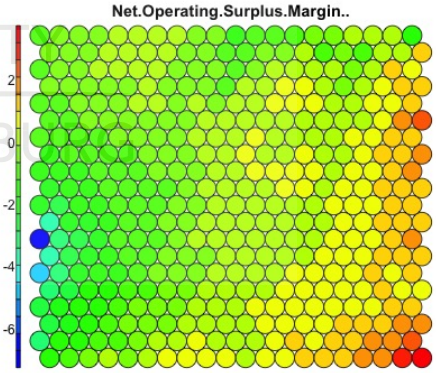
(a) Trained SOM. K-means clustering (with 10 clusters) was applied to the SOM nodes. The labels of the resultant clusters are shown in each node. For example, the green cluster is labelled as cluster 1. Table 3.4 presents the breakdown by audit opinion for each cluster.



(b) Debt to total operating revenue feature overlaid on the trained SOM in Figure 3.3(a). Nodes with a high debt to total operating revenue ratio are in the bottom left corner. These nodes correspond to clusters 2, 5 and 9 in Figure 3.3(a).



(c) Current ratio overlaid on the trained SOM in Figure 3.3(a). Nodes with a high current ratio are in the top right corner. These nodes correspond with clusters 8 and 10 in Figure 3.3(a).



(d) Net operating surplus margin feature overlaid on the trained SOM in Figure 3.3(a). Nodes with a high net operating surplus margin ratio are on the right of the figure. These nodes correspond broadly with cluster 7 in Figure 3.3(a).

Figure 3.3: Clustering results from the SOM applied to the South African municipal financial statement audit outcome dataset.

Table 3.3: Five number summary of the thirteen financial ratios in the South African municipal financial statement audit opinion dataset. Note that mil represents a million. Q1 and Q3 are the lower and upper quartiles.

Ratio	Min	Q1	Median	Q3	Max
1	-2046.45	11.97	19.43	35.49	6640.18
2	-3.81	11.22	17.85	25.36	77.66
3	-15.34	3.67	4.94	6.58	159.27
4	-1.86	0.00	0.83	1.91	170.67
5	-0.09	0.28	0.43	0.63	2.64
6	-0.88	0.56	1.16	2.19	23.06
7	-0.31	0.207	0.98	2.33	13.97
8	-147.32	-7.08	4.71	15.34	81.07
9	-67.61	26.31	32.51	41.17	159.98
10	-11.31	0.00	0.97	4.91	51.83
11	7.87	98.14	99.85	100.00	103.54
12	-114100 mil	0	0	83	1436 mil
13	-555400 mil	0	0	21	14970 mil

three clusters in the table) consist of predominantly qualified / fraudulent instances. Looking at Figure 3.3(a), we see that the mostly non-fraudulent clusters (purple, white and green) are positioned together in the top-right corner of the SOM, while the mostly fraudulent clusters (grey, brown and blue) are positioned together in the bottom-left corner of the SOM. The remaining clusters between these two extremes have a roughly 50-50 split of non-fraudulent to fraudulent classes, and so it is not clear whether these clusters are of fraudulent entities or not.

Figures 3.3(b)-3.3(d) display the results of overlaying the SOM with the financial ratio features, being the *debt to total operating revenue*, *current ratio* and the *net operating surplus margin*. In these feature maps, red neurons correspond with the highest values of the feature, while blue neurons correspond with the lowest values of the feature. For example, we see in Figure 3.3(c) that the instances that have the highest values of *current ratio* are positioned in the top right corner of the SOM, which corresponds with cluster

Table 3.4: The ten clusters on the SOM in Figure 3.3(a) and associated breakdown by audit opinion. UQ = Unqualified, Q = qualified, D = disclaimer and A = adverse. The colours of the clusters correspond to those in Figure 3.3(a).

Cluster (colour)	UQ	Q	D	A	**Total**
1 (green)	128	44	7	4	**183**
2 (blue)	27	55	46	1	**129**
3 (red)	251	130	52	11	**444**
4 (yellow)	129	51	45	6	**231**
5 (gray)	7	8	8	0	**23**
6 (pink)	86	60	32	5	**183**
7 (orange)	155	84	27	10	**276**
8 (purple)	12	1	0	0	**13**
9 (brown)	4	3	8	0	**15**
10 (white)	57	5	1	0	**63**
Total	**856**	**441**	**226**	**37**	**1 560**

Table 3.5: Percentage distribution between unqualified and not unqualified (i.e. qualified, disclaimer and adverse) in each of the ten clusters on the SOM in Figure 3.3(a). The colours of the clusters correspond to those in Figure 3.3(a).

Cluster (colour)	Unqualified	Not-unqualified
8 (purple)	**92%**	8%
10 (white)	**90%**	10%
1 (green)	**70%**	30%
3 (red)	**57%**	43%
4 (yellow)	**56%**	44%
7 (orange)	**56%**	44%
6 (pink)	47%	**53%**
5 (gray)	30%	**70%**
9 (brown)	27%	**73%**
2 (blue)	21%	**79%**

8 (the purple cluster) in Figure 3.3(a). Note that overlying the other ten financial ratios outlined in Section 3.3.4 did not produce clear partitions of the clusters, and we thus do not include the results.

Looking simultaneously at Figure 3.3(a) and Figures 3.3(b) - 3.3(d), it is clear that high values of the current ratio are associated with clusters 8 and 10, which are non-fraudulent classes, while high values of the debt to total operating revenue ratio are associated with clusters 2, 5 and 9, which are the fraudulent clusters. High values of the net operating margin surplus ratio are associated with cluster 7, which is a moderately (with 57%) non-fraudulent cluster. On the other hand, the lowest values of the net operating margin (blue nodes) fall into cluster 9 (brown cluster), which is primarily fraudulent.

These results make intuitive sense because a high current ratio means that the entity has more current assets than current liabilities, meaning that it is in an excellent financial position and less likely to be fraudulent. On the other hand, a high debt to total operating revenue ratio means that the entity has too much debt compared to the revenue that it is generating, which makes it more likely to act fraudulently. In addition, entities with high net operating surplus margins are profitable and less likely to be fraudulent. In contrast, entities with meager and negative net operating surplus margins are more likely to act fraudulently. These results are in line with what has been observed for listed entities in previous studies [128, 58].

3.4 Performance Metrics

We now present the performance metrics used to measure the performance of the algorithms proposed in this thesis. The performance metrics used are the acceptance rate, the multivariate ESS, the multivariate ESS normalised by the execution time, as well predictive performance on unseen data. We also assess the convergence of the proposed MCMC methods using the potential scale reduction factor metric. The acceptance rate metric measures the number of generated samples that are accepted in the MH acceptance step of the algorithm. The higher the number of accepted samples for the same step size, the more preferable the method. We discuss the remaining metrics in more

detail in the following sections.

3.4.1 Effective sample size

The ESS metric is a commonly used metric for assessing the sampling efficiency of an MCMC algorithm. It indicates the number of effectively uncorrelated samples out of the total number of generated samples [140]. The larger the ESS, the better the performance of the MCMC method. The ESS normalised by execution time metric takes into account the computational resources required to generate the samples and penalises MCMC methods that require more computational resources to generate the same number of uncorrelated samples. The larger this metric, the better the efficiency of the algorithm.

This thesis employs the multivariate ESS metric developed by Vats *et al.* [160] instead of the minimum univariate ESS metric typically used in analysing MCMC results. The minimum univariate ESS measure is not able to capture the correlations between the different parameter dimensions, while the multivariate ESS metric can incorporate this information [112, 160, 53, 113]. The minimum univariate ESS calculation results in the estimate of the ESS being dominated by the parameter dimensions that mix the slowest and ignore all other dimensions [160, 112]. The multivariate ESS is calculated as:

$$\text{mESS} = N \times \left(\frac{|\Lambda|}{|\Sigma|}\right)^{\frac{1}{D}} \tag{3.10}$$

where N is the number of generated samples, D is the number of parameters, $|\Lambda|$ is the determinant of the sample covariance matrix and $|\Sigma|$ is the determinant of the estimate of the Markov chain standard error. When $D = 1$, mESS is equivalent to the univariate ESS measure [160]. Note that when there are no correlations in the chain, we have that $|\Lambda| = |\Sigma|$ and $\text{mESS} = N$.

We now address the ESS calculation for Markov chains that have been re-weighted via importance sampling, such is the case for the shadow HMC algorithms considered in this thesis [140, 68, 112, 99]. For N samples re-weighted by importance sampling, the common approach is to use the approximation by Kish [83, 68] given by

$$\text{ESS}_{IMP} = \frac{1}{\left(\sum_{j=1}^{N} \bar{b}_j^2\right)} \tag{3.11}$$

where $\bar{b}_j = b_j / \sum_{k=1}^{N} b_k$. This accounts for the possible non-uniformity in the importance sampling weights. In order to account for both the effects of sample auto-correlation and re-weighting via importance sampling, we approximate ESS under importance sampling by taking directions from Heide *et al.* [68] and using:

$$\text{ESS} := \frac{\text{ESS}_{IMP}}{N} \times \text{mESS} = \frac{1}{\left(\sum_{j=1}^{N} \bar{b}_j^2\right)} \times \left(\frac{|\Lambda|}{|\Sigma|}\right)^{\frac{1}{D}} \tag{3.12}$$

3.4.2 Convergence analysis

The $\hat{R}$ diagnostic of Gelman and Rubin [51] is a popular method for establishing the convergence of MCMC chains [147]. This diagnostic relies on running multiple chains $\{X_{i0}, X_{i1}, ..., X_{i(N-1)}\}$ for $i \in \{1, 2, 3, ..., m\}$ staring at various initial states with m being the number of chains and N being the sample size. Using these parallel chains, two estimators of the variance can be constructed. The estimators are the between-the-chain variance estimate and the within-the-chain variance. When the chain has converged, the ratio of these two estimators should be one. The $\hat{R}$ metric, which is formally known as the potential scale reduction factor, is defined as:

$$\hat{R} = \frac{\hat{V}}{W} \tag{3.13}$$

where

$$W = \sum_{i=1}^{m} \sum_{j=0}^{N-1} \frac{\left(X_{ij} - \bar{X}_{i.}\right)^2}{m(N-1)} \tag{3.14}$$

is the within-chain variance estimate and $\hat{V} = \frac{N-1}{N} W + \frac{B}{N}$ is the pooled variance estimate which incorporates the between-chains

$$B = \sum_{j=0}^{N-1} \frac{\left(\bar{X}_{i.} - \bar{X}_{..}\right)^2}{m-1} \tag{3.15}$$

and within-chain W variance estimates, with $\bar{X}_{i.}$ and $\bar{X}_{..}$ being the i^{th} chain mean and overall mean respectively for $i \in \{1, 2, 3, ..., m\}$. Values larger than the convergence threshold of 1.05 for the $\hat{R}$ metric indicate divergence of the chain [26, 51]. In this thesis, we asses the convergence of the chains by computing the *maximum* $\hat{R}$ metric over each of the parameter dimensions for the given target.

3.4.3 Predictive performance on unseen data

Examples of performance measures used for classification problems is shown in Table 3.7 where TP is the true positive, FN is the false negative, FP is the false positive and TN is the true negative. Other examples of performance measures include Receiver Operating Curve (ROC) as well as Area Under The Receiver Operating Curve (AUC). ROC plots the true positives from the model on the y-axis against the false positives on the x-axis [109]. AUC is the area under the ROC, and represents the average miss-classifications rate. AUC is useful as a performance measure when the costs of classification are unknown [22, 49, 109], such as when modelling FSF.

Table 3.6: Common classification performance measures.

accuracy $= \frac{TP+TN}{TP+FP+FN+TN}$	sensitivity $= \frac{TP}{TN+FN}$	specificity $= \frac{TN}{TN+FP}$
precision $= \frac{TP}{TP+FP}$	recall $= \frac{TP}{TP+FN}$	F-Measure $= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

Examples of performance metrics used in regression problems are the coefficient of determination (R^2), Mean Absolute Error (MAE), and the Mean Square Error (MSE) and are shown in Table 3.6 [99]. In this table, we have that K_i are the observations with average $\bar{K}$, M is the number of observations and P_i are the predictions. MSE and MAE are similar in that they measure the difference between the observations and the predictions, while the R^2 measures the percentage of the variability in the target variable that is explained by the model. Models that have low values for MSE and MAE are preferable, while models that have a high R^2 measure are preferable.

Table 3.7: Common regression performance measures.

MSE $= \frac{1}{M}\sum_{i=1}^{M}\left(K_i - P_i\right)^2$	MAE $= \frac{1}{M}\sum_{i=1}^{M}\left\|K_i - P_i\right\|$	$R^2 = 1 - \frac{\sum_{i=1}^{M}\left(K_i - P_i\right)^2}{\sum_{i=1}^{M}\left(K_i - \bar{K}\right)^2}$

3.5 Algorithm Parameter Tuning

As mentioned in Section 2.5, the matrix $\mathbf{G}$ in the MHMC method provides an extra degree of freedom which typically results in better sampling behavior than HMC [112, 156, 25]. It is not immediately clear how this matrix should be set - this is still an open area of research [112, 156, 26]. In this thesis, we take direction from the inventors [156] of the method and select only a few dimensions to be influenced by the magnetic field. In particular, $\mathbf{G}$ was set such that $\mathbf{G_{1i}} = g$, $\mathbf{G_{i1}} = -g$ and zero elsewhere where $g = 0.2$ for the BLR datasets, and $g = 0.1$ for all the other targets.

Table 3.8: Step size and trajectory length parameters used for HMC and MHMC in this thesis. These methods serve as the baselines against which the novel methods presented in this thesis are compared. Five thousand samples were used to tune the step size for the given trajectory length using primal-dual averaging. The target acceptance rate was set to 80%.

Problem	L	**HMC**	**MHMC**
Banana	15	0.1209	0.1179
Neal with $D = 25$	50	0.3823	0.0507
Gaussian with $D = 10$	15	0.3491	0.1414
Gaussian with $D = 50$	15	0.1491	0.1219
Bitcoin	15	0.0021	0.0021
S&P 500	15	0.0028	0.0027
USDZAR	15	0.0003	0.0003
Pima	50	0.1062	0.0300
Heart	50	0.1595	0.0241
Australian	50	0.1060	0.0300
German	50	0.0572	0.0413
Power	25	0.0281	0.0275
Airfoil	25	0.0547	0.0497
Concrete	25	0.0761	0.0634
Protein	25	0.0793	0.0691

These settings mean that the choice of $\mathbf{G}$ is not necessarily the optimal choice for

all the target distributions considered, but was sufficient for our purposes as this basic setting still leads to good performance on the algorithms that we propose in this thesis. Tuning $\mathbf{G}$ for each target posterior should result in improved performance compared to the results presented in this manuscript. An alternative approach to the selection of $\mathbf{G}$ would have been to follow [26] and selecting $\mathbf{G}$ to be a random antisymmetric matrix. It is not immediately clear if the approach of [26] is necessary optimal, and we plan to explore this approach in future work.

In this thesis, unless otherwise stated, we tune the step size ϵ parameter for each problem for a given fixed trajectory length L parameter for the HMC and MHMC methods, with $\mathbf{G}$ being as described above. The methods we construct on top of these two algorithms then use the same step size as these two base methods. The step size is tuned to target an acceptance rate of **80%** using the primal-dual averaging methodology [73]. The trajectory lengths used vary across the different targets, with the final step sizes and trajectory lengths used for the various problems presented in Table 3.8.

All the experiments in this thesis were conducted on a machine with a 64bit CPU using PyTorch. We are in the process of making the PyTorch code used in this thesis open source.

3.6 Conclusion

This chapter outlined the target distributions considered in this thesis. Various combinations of these target distributions will be used throughout the thesis. We also introduced a new public interest dataset being the South African municipal financial statement audit outcome dataset. The preliminary analysis performed on this dataset using a SOM showed that financial ratios are useful in modelling audit outcomes of local government entities. Furthermore, the important financial ratios were identified as the *current ratio*, *debt to total operating revenue* as well as the *net surplus profit margin*.

In the next chapter, we present the novel QIMHMC algorithm, which employs a random mass matrix $\mathbf{M}$ for the auxiliary momentum variable $\mathbf{p}$ in MHMC to enhance the performance of MHMC.

Chapter 4

Quantum-Inspired Magnetic Hamiltonian Monte Carlo

4.1 Introduction

Liu and Zhang [91] have recently shown in their work on QIHMC that making use of the energy-time uncertainty relation from quantum mechanics, one can devise an extension to HMC by allowing the mass matrix of the auxiliary momentum variable to be random with a probability distribution instead of being fixed. Furthermore, MHMC has been proposed as an enhancement to HMC by adding a magnetic field to HMC which results in non-canonical dynamics associated with the movement of a particle under a magnetic field. In this chapter, we utilise the non-canonical dynamics of MHMC while allowing the mass matrix to be random to create the novel QIMHMC algorithm, which is shown to converge to the correct steady-state distribution. Empirical results on a broad class of target posterior distributions and various performance metrics show that the proposed method produces better sampling performance than HMC, MHMC and QIHMC. The work in this chapter was published in the following international journal article:

> **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Quantum-Inspired Magnetic Hamiltonian Monte Carlo*. PloS One, vol. 16, no. 10, pp. e0258277.

4.2 Proposed Algorithm

One of the parameters of HMC and MHMC that needs to be set by the user is the mass matrix $\mathbf{M}$ of the auxiliary momentum variable. This mass matrix is typically set to equal the identity matrix $\mathbf{I}$ [156, 53, 122, 127]. Although this typically produces good results in practice, it is not necessarily the optimal choice across all possible target distributions. An approach to address this drawback would be to set the mass matrix to depend on the target distribution's local geometry or make the mass matrix a stochastic process with a user-specified distribution.

In RMHMC, the mass matrix is the Hessian of the negative log-density, which is the fisher information metric [53, 16]. This allows the method to take into account the local curvature of the target and has been shown to enhance the sampling performance of HMC, especially for ill-conditioned target distributions [53, 35, 68]. On the other hand, QIHMC [91] sets the mass matrix to be a stochastic process and varies for each sample generated. This is motivated by the energy-time uncertainty relation from quantum mechanics, which allows a particle's mass to be stochastic rather than fixed, as is the case with classical particles [91].

QIHMC has been shown to improve the sampling performance when sampling from a broad class of distributions which occur in sparse modeling via bridge regression, image denoising, and BNN pruning [91, 113]. This is particularly important for spiky and multi-modal distributions where HMC is inefficient [91, 113]. The use of a random mass matrix is yet to be considered for MHMC in the literature. Given that MHMC is closely related to HMC and outperforms HMC for a well-chosen magnetic field, one would expect that making the mass matrix random in MHMC would result in improved sampling performance when compared to MHMC with a fixed mass.

In this chapter, we set the mass matrix $\mathbf{M}$ in MHMC in Algorithm 3 to be random with distribution $\mathbf{P}_{\mathbf{M}}(\mathbf{M})$ to create the novel QIMHMC method. This proposed algorithm has simiarly dynamics to MHMC in equation (2.22), with the exception that the mass matrix is random and is re-sampled before generating the auxiliary momentum variable. The algorithmic description of QIMHMC is presented in Algorithm 9. QIMHMC only differs from MHMC by the addition of line 3 in Algorithm 9 and a modified integrator in line 5, with every other step of the algorithm being the same as that

of MHMC in Algorithm 3.

Note that the algorithm for QIHMC used in this chapter is the one outlined in Algorithm 4.

Algorithm 9 Quantum-Inspired Magnetic Hamiltonian Monte Carlo Algorithm

Input: N, ϵ, L, w_{init}, $H(w,p)$, $\mathbf{G}$, $\mathbf{P_M}(\mathbf{M})$

Output: $(w)_{m=0}^{N}$

1: $w_0 \leftarrow w_{\text{init}}$

2: **for** $m \rightarrow 1$ **to** N **do**

3: $\mathbf{M} \sim \mathbf{P_M}(\mathbf{M})$ $\leftarrow$ **re-sample mass matrix.**

4: $p_{m-1} \sim \mathcal{N}(0, \mathbf{M})$

5: p_m, w_m = **Integrator**$(p_{m-1}, w_{m-1}, \epsilon, L, H, \mathbf{G}, \mathbf{M})$ in equation (4.1)

Remaining steps proceed as in MHMC in Algorithm 3

6: **end for**

It is worth noting that we refer to the proposed algorithm as Quantum-Inspired Magnetic Hamiltonian Monte Carlo as it utilises a random mass matrix, which is consistent with the behaviour of quantum particles. A particle can have a random mass with a distribution in quantum mechanics, while in classical mechanics, a particle has a fixed mass. The classical version of the proposed algorithm is the Magnetic Hamiltonian Monte Carlo method. When a random mass is utilised as inspired by quantum particles, the result is Quantum-Inspired Magnetic Hamiltonian Monte Carlo. Furthermore, this naming convention is consistent with that used by Liu and Zhang [91] for Hamiltonian Monte Carlo. Their work differs from ours in that the quantum particle is now subjected to a magnetic field.

In Theorem 4.2.1, we extend equation (2.22) of Tripuraneni *et al.* [156] to allow for the presence of a mass matrix that is not equal to the identity, that is for the case $\mathbf{M} \neq \mathbf{I}$. This extension is required for the implementation of the new QIMHMC method that we are proposing. This is because the mass matrix $\mathbf{M}$ will be changing for each new position sample generated, and this new $\mathbf{M}$ must be incorporated in the integration scheme.

Theorem 4.2.1. *The leapfrog-like numerical integration scheme for MHMC in the presence of a mass matrix* $\mathbf{M}$ *that is not equal to the identity matrix* $\mathbf{I}$ *is:*

$$
\begin{aligned}
\mathbf{p}_{t+\frac{\epsilon}{2}} &= \mathbf{p}_t + \frac{\epsilon}{2}\frac{\partial H\left(\mathbf{w}_t, \mathbf{p}_t\right)}{\partial \mathbf{w}} \\
\mathbf{w}_{t+\epsilon} &= \mathbf{w}_t + \mathbf{G}^{-1}\left(\exp\left(\mathbf{G}\mathbf{M}^{-1}\epsilon\right) - \mathbf{I}\right)\mathbf{p}_{t+\frac{\epsilon}{2}} \\
\mathbf{p}_{t+\frac{\epsilon}{2}} &= \exp\left(\mathbf{G}\mathbf{M}^{-1}\epsilon\right)\mathbf{p}_{t+\frac{\epsilon}{2}} \\
\mathbf{p}_{t+\epsilon} &= \mathbf{p}_{t+\frac{\epsilon}{2}} + \frac{\epsilon}{2}\frac{\partial H\left(\mathbf{w}_{t+\epsilon}, \mathbf{p}_{t+\frac{\epsilon}{2}}\right)}{\partial \mathbf{w}}.
\end{aligned}
\tag{4.1}
$$

Proof. The Hamiltonian in equation (2.13) can be re-expressed as:

$$
\begin{aligned}
\mathrm{H}(\mathbf{w}, \mathbf{p}) &= \frac{\mathrm{U}(\mathbf{w})}{2} + \mathrm{K}(\mathbf{p}) + \frac{\mathrm{U}(\mathbf{w})}{2} \\
&= \underbrace{\frac{\mathrm{U}(\mathbf{w})}{2}}_{H_1(\mathbf{w})} + \underbrace{\frac{1}{2}\log\left((2\pi)^D|\mathbf{M}|\right) + \frac{\mathbf{p}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{p}}{2}}_{H_2(\mathbf{p})} + \underbrace{\frac{\mathrm{U}(\mathbf{w})}{2}}_{H_1(\mathbf{w})} \\
\mathrm{H}(\mathbf{w}, \mathbf{p}) &= H_1(\mathbf{w}) + H_2(\mathbf{p}) + H_1(\mathbf{w})
\end{aligned}
\tag{4.2}
$$

where each of the sub flows $H_1(\mathbf{w})$ and $H_2(\mathbf{p})$ can be integrated exactly as follows:

$$
\begin{aligned}
\frac{d}{dt}\begin{bmatrix}\mathbf{w}\\ \mathbf{p}\end{bmatrix} &= \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{G}\end{bmatrix}\begin{bmatrix}\nabla_w H_1(\mathbf{w})\\ \nabla_p H_1(\mathbf{w})\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{G}\end{bmatrix}\begin{bmatrix}\frac{\mathrm{U}(\mathbf{w})}{2}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ -\frac{\mathrm{U}(\mathbf{w})}{2}\end{bmatrix} \\
\implies d\mathbf{p} &= -\frac{dt}{2}\mathrm{U}(\mathbf{w}) \\
\mathbf{p} &= \mathbf{p} - \frac{\epsilon}{2}\mathrm{U}(\mathbf{w}) \\
\implies \Phi_{\epsilon, H_1(\mathbf{w})}\begin{bmatrix}\mathbf{w}\\ \mathbf{p}\end{bmatrix} &= \begin{bmatrix}\mathbf{w}\\ \mathbf{p} - \frac{\epsilon}{2}\mathrm{U}(\mathbf{w})\end{bmatrix}
\end{aligned}
\tag{4.3}
$$

and

$$\begin{aligned}
\frac{d}{dt}\begin{bmatrix}\mathbf{w}\\ \mathbf{p}\end{bmatrix} &= \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{G}\end{bmatrix}\begin{bmatrix}\nabla_w H_2(\mathbf{p})\\ \nabla_p H_2(\mathbf{p})\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{G}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \mathbf{M^{-1}p}\end{bmatrix} = \begin{bmatrix}\mathbf{M^{-1}p}\\ \mathbf{GM^{-1}p}\end{bmatrix}\\
\implies d\mathbf{p} &= \mathbf{GM^{-1}}\mathbf{p}dt\\
\mathbf{p} &= \exp\left(\mathbf{GM^{-1}}t\right)\mathbf{p}\\
d\mathbf{w} &= \mathbf{M^{-1}}\exp\left(\mathbf{GM^{-1}}t\right)\mathbf{p}dt \qquad (4.4)\\
\implies \mathbf{w} &= \mathbf{w} + \mathbf{M^{-1}}(\mathbf{GM^{-1}})^{-1}\left(\exp\left(\mathbf{GM^{-1}}t\right) - \mathbf{I}\right)\mathbf{p}\\
\mathbf{w} &= \mathbf{w} + \mathbf{G}^{-1}\left(\exp\left(\mathbf{GM^{-1}}t\right) - \mathbf{I}\right)\mathbf{p}\\
\implies \Phi_{\epsilon,H_2(\mathbf{p})}\begin{bmatrix}\mathbf{w}\\ \mathbf{p}\end{bmatrix} &= \begin{bmatrix}\mathbf{w} + \mathbf{G}^{-1}\left(\exp\left(\mathbf{GM^{-1}}\epsilon\right) - \mathbf{I}\right)\mathbf{p}\\ \exp\left(\mathbf{GM^{-1}}\epsilon\right)\mathbf{p}\end{bmatrix}
\end{aligned}$$

Equation (4.1) is then generated by the map $\Phi_{\epsilon,H_{(\mathbf{w},\mathbf{p})}} = \Phi_{\epsilon,H_1(\mathbf{w})} \circ \Phi_{\epsilon,H_2(\mathbf{p})} \circ \Phi_{\epsilon,H_1(\mathbf{w})}$. □

In the QIMHMC algorithm, we introduced another source of randomness in the form of a random mass matrix into MHMC. We thus need to show that the proposed algorithm converges to the correct steady-state distribution. This is provided in Theorem 4.2.2, which guarantees that the proposed algorithm produces the correct steady-state distribution.

Theorem 4.2.2. *Consider continuous-time non-canonical Hamiltonian dynamics with a deterministic time-varying positive-definite mass matrix* $\mathbf{M}(t)$ *in equation* (4.1)*. The marginal density* $\pi_{\mathbf{w}}(\mathbf{w}) \propto \exp(-U(\mathbf{w}))$ *is a unique steady state distribution in the* $\mathbf{w}$ *space if momentum re-sampling steps* $p_{\mathbf{p}}(\mathbf{p}) \propto \exp\left(\frac{\mathbf{p}^{\mathrm{T}}\mathbf{M}(t)^{-1}\mathbf{p}}{2}\right)$ *are included.*

Proof. Following the approach of [91] for QIHMC, we consider the joint distribution of $(\mathbf{w}, \mathbf{p}, \mathbf{M})$ given by $\pi(\mathbf{w}, \mathbf{p}, \mathbf{M})$. Here we have dropped the explicit dependence of $\mathbf{M}$ on t because $\mathbf{M}(t)$ obeys the mass distribution $\mathbf{P}_{\mathbf{M}}(\mathbf{M})$ for all t. Employing Bayes theorem

we have that:

$$\begin{aligned}
\pi\left(\mathbf{w},\mathbf{p},\mathbf{M}\right) &= p\left(\mathbf{w},\mathbf{p}|\mathbf{M}\right)\mathbf{P}_{\mathbf{M}}(\mathbf{M}) \\
\pi\left(\mathbf{w},\mathbf{p},\mathbf{M}\right) &\propto \exp(-U(\mathbf{w}))\exp\left(\frac{\mathbf{p}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{p}}{2}\right) \\
\implies \pi\left(\mathbf{w}\right) &= \int_{\mathbf{p}}\int_{\mathbf{M}} \pi\left(\mathbf{w},\mathbf{p},\mathbf{M}\right) d\mathbf{w}d\mathbf{M} \\
\pi\left(\mathbf{w}\right) &\propto \exp(-U(\mathbf{w}))
\end{aligned} \tag{4.5}$$

which means that the marginal steady state distribution $\pi\left(\mathbf{w}\right)$ is the correct posterior distribution. □

An aspect that we are yet to address is which distribution $\mathbf{P}_{\mathbf{M}}(\mathbf{M})$ should be used for the mass matrix. This is still an open area of research [91, 113]. In this work, we consider the simple case where $\mathbf{M}$ is a diagonal matrix with the entries being sampled from a log-normal distribution where the mean is zero and the standard deviation, which we refer to as the volatility of volatility or vol-of-vol for short, is equal to a tunable parameter $\beta \in \mathbb{R}^{\geq 0}$. That is, for each $m_{ii} \in \mathbf{M} \quad \forall \quad i \in \{1, 2, ..., D\}$ is generated as $\ln m_{ii} \sim \mathcal{N}(0, \beta^2)$ and $m_{ij} = 0$ for $i \neq j$. We present the sensitivity analysis to the chosen value of β in Section 4.3.2.

4.3 Experiment Description

In this section, we outline the settings used for the experiments, the performance metrics, and we present the sensitivity analysis for the vol-of-vol parameter β in QIMHMC.

4.3.1 Experiment settings

In all our experiments, we compare the performance of QIMHMC to HMC, MHMC and QIHMC using the multivariate ESS and the ESS normalised by execution time metrics presented in Section 3.4.1. We further assess the convergence behaviour of the MCMC methods using the $\hat{R}$ metric described in Section 3.4.2. The targets we consider in this chapter are the Banana shaped distribution, the multivariate Gaussian distributions with

$D \in \{10, 50\}$ and BLR on the Pima, Australian and German credit datasets outlined in Table 3.2.

The vol-of-vol parameter β that we use depends on the particular target density. The sensitivity analysis for β is presented in Section 4.3.2. We set β to 0.1 for the Banana shaped distribution and Australian credit dataset, while β was set to 0.3 for the other targets. The trajectory length and step size parameters for the MCMC methods considered in this chapter were set to the values outlined in Table 3.8, with QIHMC and QIMHMC using the same step size as HMC and MHMC respectively. Ten independent chains were run for each method on each target distribution. Three thousand samples were generated for each target, with the first 1 000 samples discarded as burn-in. This sample size and burn-in period were sufficient for all the algorithms to converge on all the targets. It is worth highlighting that we set the mass matrix $\mathbf{M}$ to the identity matrix for the HMC and MHMC methods. This setting for $\mathbf{M}$ is what is commonly used in practice [17, 156, 59, 164].

4.3.2 Sensitivity to the vol-of-vol parameter

In this section, we present the sensitivity analysis for the chosen vol-of-vol parameter β. We considered values of $\beta \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1.0\}$, with all the other settings being the same as those outlined in Section 4.3.1. It is worth highlighting that when $\beta = 0$, QIMHMC is equivalent to MHMC and QIHMC is equivalent to HMC. The results of the analysis are presented in Figure 4.1. The results show that the ESS has a tendency of decreasing for both QIHMC and QIMHMC with increasing β on both the Australian and German credit datasets. In addition, the QIMHMC algorithm is able to maintain a high acceptance rate as the value of β is increased, while the acceptance rate of QIHMC decays to zero at a faster rate.

The results also indicate that smaller values of β are preferable, with the optimal value of the Australian credit dataset being 0.1 and around 0.5 for the German credit dataset. Furthermore, QIMHMC outperforms QIHMC for all values of β on both an ESS and normalised ESS basis, showing the robust results that can be obtained from QIMHMC.

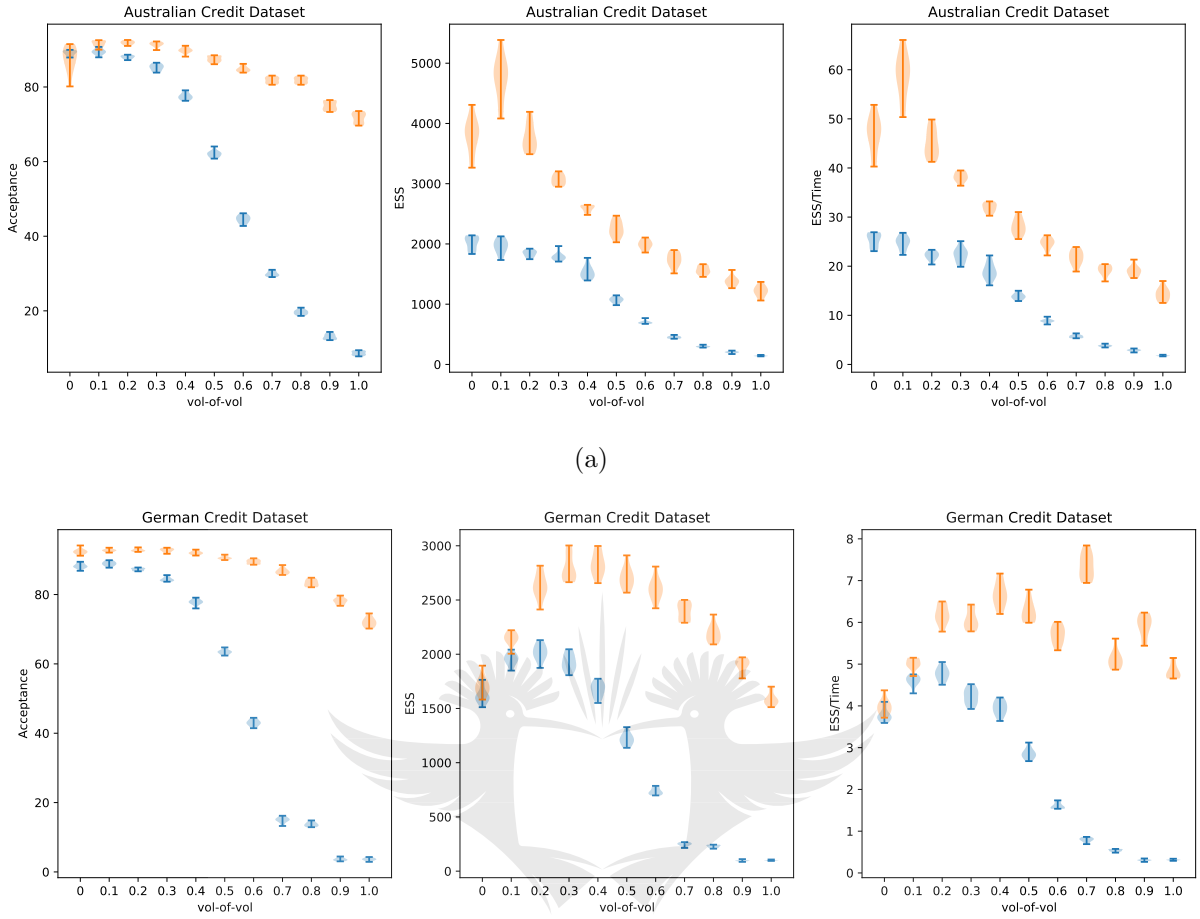


Figure 4.1: Acceptance rates, ESS and ESS/Time for ten runs of QIHMC (blue) and QIMHMC (orange) on the a) Australian and b) German credit datasets with varying choices of the vol-of-vol β parameter. The results indicate that these metrics are a decreasing function of the vol-of-vol β parameter, with QIMHMC decreasing at a slower rate than QIHMC.

4.4 Results and Discussion

We present the performance of the HMC, MHMC, QIHMC, and QIMHMC methods using different performance metrics in Figure 4.3 and Tables 4.1 to 4.3. In Figure 4.3, the plots on the first row for each dataset show the ESS, while the plots on the second row show the ESS normalised by execution time. The results are for the ten runs of each algorithm. In Tables 4.1 to 4.3, each column corresponds to the results for a particular

MCMC method. Values that are in **bold** indicate that the MCMC method outperforms the other MCMC algorithms on that particular metric. The execution time t in Figure 4.3 and Tables 4.1 to 4.3 is in seconds. The results in Tables 4.1 to 4.3 are the mean results over the ten runs for each algorithm. We use the mean values over the ten runs in Tables 4.1 to 4.3 to form our conclusions about the performance of the algorithms.

The results in Tables 4.1 to 4.3 and Figure 4.2 show that all the MCMC methods have converged as indicated by the $\hat{R}$ metric and the diagnostic trace-plots for the negative log-likelihood. It is also worth noting that the quantum-inspired algorithms consistently produce better convergence across the different targets based on the $\hat{R}$ metric, indicating that $\beta \neq 0$ improves the convergence behaviour of QIHMC and QIMHMC respectively.

The results in Figure 4.3 and Tables 4.1 to 4.3 show that the proposed QIMHMC method outperforms all the other methods across the ESS and normalised ESS performance metrics on all the targets. The outperformance also increases with increasing dimensionality of the problem, suggesting that QIMHMC is able to scale to larger models without a significant deterioration in sampling performance. It is also worth noting that MHMC sometimes outperforms QIHMC, for example on the Gaussian distribution with $D = 10$, suggesting that MHMC with an appropriately tuned magnetic field is able to outperform QIHMC.

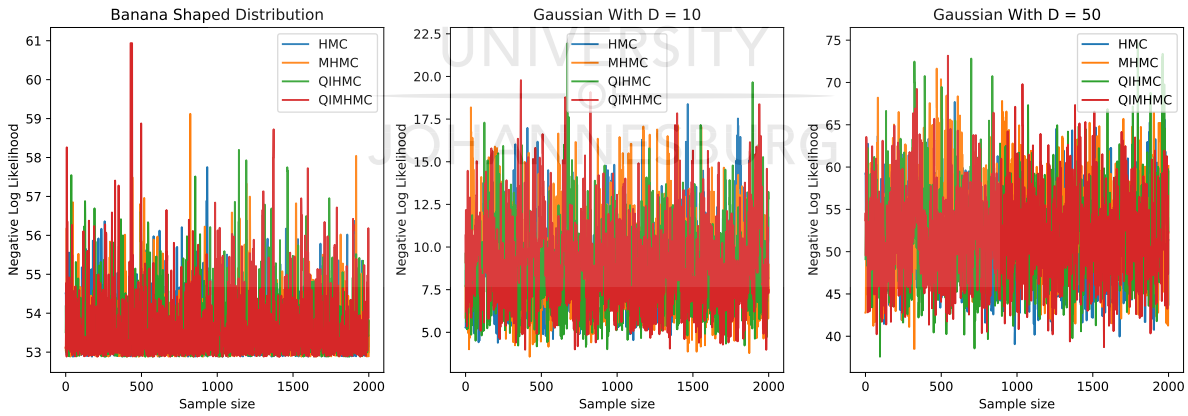


Figure 4.2: Diagnostic trace-plots of the negative log-likelihood across various targets averaged over ten runs of each method. These results show that all the MCMC methods have converged on all the targets.

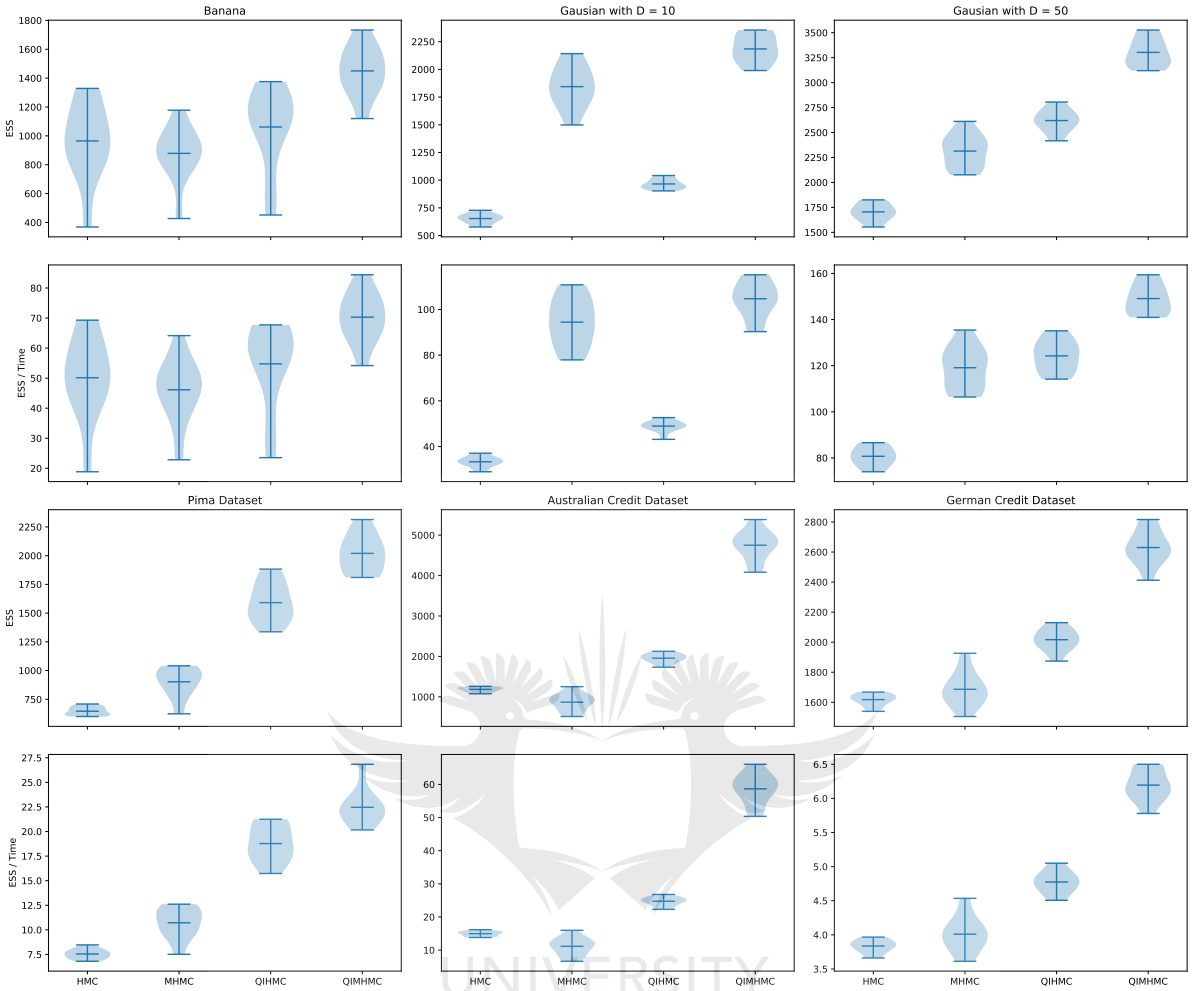


Figure 4.3: Results for the datasets over ten runs of each method. For each dataset, the plots on the first row show the multivariate effective sample size and the plots on the second row show the multivariate effective sample size normalised by execution time (in seconds). For all the plots, the larger the value the better the method. The dark horizontal line in each violin plot represents the mean value over ten runs of each algorithm.

As expected, MHMC outperforms HMC on the majority of the targets across all the metrics. This is in line with what has been previously observed [156, 26, 112]. The MHMC method underperforms HMC on the Banana and Australian target distributions, suggesting that the magnetic field chosen for these two targets might not be optimal. It is also worth noting that MHMC and HMC produce similar execution times, with HMC

Table 4.1: Banana shaped distribution results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric.

Banana shaped distribution				
Metric	HMC	MHMC	QIHMC	QIMHMC
ESS	964	878	1 061	**1 450**
t	19.24	**19.12**	19.34	20.63
ESS/t	50.1	46.1	54.8	**70.3**
$\hat{R}$ max	1.01	1.02	**1.00**	**1.00**

marginally outperforming MHMC on the majority of the targets on an execution time basis.

Table 4.2: Multivariate Gaussian distribution results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric.

Gaussian with D = 10				
Metric	HMC	MHMC	QIHMC	QIMHMC
ESS	653	1 844	965	**2 184**
t	19.6	**19.5**	19.8	20.9
ESS/t	33.3	94.5	48.9	**104.7**
$\hat{R}$ max	1.03	1.02	**1.00**	**1.00**
Gaussian with D = 50				
ESS	1 705	2 313	2 619	**3 302**
t	21.1	**19.4**	21.2	23.3
ESS/t	80.7	119.11	124.3	**141.7**
$\hat{R}$ max	1.01	1.03	**1.00**	**1.00**

We also noted that QIHMC outperforms HMC across all the targets. This confirms the results observed by Liu and Zhang [91] using different target posteriors to those

considered in this chapter. This shows the significant benefit that utilising a random mass can provide to the sampling properties of HMC based samplers. However, the real performance gains are only realised when the vol-of-vol parameter β has been appropriately tuned. Establishing an automated approach for tuning β is still an open research problem, which we aim to address in future work.

Table 4.3: Bayesian logistic regression results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric.

Pima dataset				
Metric	HMC	MHMC	QIHMC	QIMHMC
ESS	645	902	1 591	**2 019**
t	85.5	**84.2**	84.7	89.9
ESS/t	7.5	10.7	18.8	**22.5**
$\hat{R}$ max	**1.00**	1.02	**1.00**	**1.00**
Australian credit dataset				
ESS	1 184	867	1 956	**4 750**
t	79.2	**77.8**	79.0	81.0
ESS/t	15.0	11.1	24.8	**58.7**
$\hat{R}$ max	1.01	1.04	**1.00**	1.01
German credit dataset				
ESS	1 617	1 686	1 927	**2 794**
t	453.9	**449.8**	456.41	459.3
ESS/t	3.56	3.75	4.22	**6.08**
$\hat{R}$ max	1.02	1.01	**1.00**	**1.00**

4.5 Conclusion

In this chapter, we introduced the QIMHMC method, which employs a random mass matrix for the auxiliary momentum variable in the non-canonical Hamiltonian dynamics of MHMC. This results in significant sampling improvements over MHMC. The new method is compared to HMC, MHMC and QIHMC. The methods are compared on the Banana shaped distribution, multivariate Gaussian distributions, and on real-world datasets modelled using BLR.

The empirical results show that the new method outperforms all the other methods on a time-normalised ESS basis across all the targets. Furthermore, QIHMC outperforms HMC as expected. This shows the significant benefit provided by using a random mass matrix for the momenta to the sampling properties of HMC based samplers in general.

A limitation of the method is the need to tune the vol-of-vol parameter. Although typically smaller values of the parameter improve the ESSs, a more robust approach to the selection of the parameter is still required. This work can be improved by establishing a heuristic or an automated approach to tune the vol-of-vol parameter. In addition, the tuning of the magnetic component could also be of interest as the outperformance of MHMC over HMC, and consequently, the outperformance of QIMHMC over QIHMC, depends on the chosen magnetic field. We also plan to incorporate a random mass matrix in S2HMC in future work.

In the following chapter, we extend MHMC and S2HMC by incorporating partial refreshment of the auxiliary momentum variable into MHMC and S2HMC respectively.

Chapter 5

Partial Momentum Refreshment

5.1 Introduction

In the preceding chapter, we assumed that the auxiliary momentum variable used to generate each parameter sample is fully regenerated at each step. This is not always the optimal approach of treating the momenta [74, 68, 140]. In their work on generalised HMC, Horowitz [74] showed that partially updating the momentum, and thus retaining some of the past dynamics, can improve the sampling performance of HMC and allows one to make the trajectory length equal to one while keeping the auto-correlations roughly the same. This concept has also been extensively utilised in the context of sampling from integrator-dependent shadow Hamiltonians in which the Hamiltonian is not separable [4, 68, 77, 140]. The use of partial momentum refreshment is yet to be considered for MHMC and S2HMC in the literature. In this chapter, we present novel algorithms based on MHMC and S2HMC algorithms which employ partial momentum refreshment to improve the sampling performance of MHMC and S2HMC respectively. The results across various targets and using different performance metrics show that the proposed algorithms enhance the sampling performance of the original algorithms. The results also show the overall utility of incorporating partial momentum update in Hamiltonian dynamics-based samplers. The material in this chapter has been published in the following two international journal articles:

- **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Magnetic Hamiltonian*

Monte Carlo With Partial Momentum Refreshment. IEEE Access, vol. 9, pp. 108009-108016.

- **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Utilising Partial Momentum Refreshment in Separable Shadow Hamiltonian Hybrid Monte Carlo.* IEEE Access, vol. 9, pp. 151235-151244.

5.2 Proposed Partial Momentum Retention Algorithms

The sampling process for HMC, MHMC and S2HMC follows a Gibbs sampling scheme where the auxiliary momentum variable is generated first, after which the position vector is generated based on the recently drawn momenta. The momenta are fully regenerated each time a new sample is drawn, with the old momenta being discarded. This is not always the optimal approach of treating the momenta across all possible target posterior distributions [74, 68, 140, 29].

Horowitz [74] was amongst the first to observe this and employed a partial momentum update to HMC and found that it significantly improved the performance of HMC. That is, keeping some of the dynamics of the chain improved performance without harming the legitimacy of the Monte Carlo method [74, 75, 42]. The concept of partial momentum refreshment has been further developed in the context of sampling from integrator dependent shadow Hamiltonians [4, 68, 77, 140]. Akhmatskaya *et al.* [4, 140] apply partial momentum refreshment to shadow Hamiltonians on the Euclidean space, while Heide *et al.* [68] employ it when sampling from shadow Hamiltonians on the Riemannian manifold. The use of partial momentum refreshment is yet to be considered for MHMC and S2HMC in the literature. Given that MHMC is closely related to HMC, one would expect that employing partial momentum refreshment in MHMC would result in the same or better sampling performance as observed by Horowitz [74] on HMC. Similarly, utilising partial momentum refreshment in S2HMC should improve the sampling behaviour of S2HMC.

In this chapter, we combine the non-canonical dynamics of MHMC with partial mo-

mentum refreshment to create the PMHMC algorithm, which converges to the target distribution. We also combine the separable Hamiltonian in S2HMC with partial momentum refreshment to create the PS2HMC algorithm, which also converges to the correct stationary distribution. The performance of the proposed methods is compared against HMC, MHMC, S2HMC and PHMC respectively.

Algorithm 10 Magnetic Hamiltonian Monte Carlo with Partial Momentum Refreshment Algorithm

Input: L, ϵ, ρ, N, $\mathbf{G}$ and $(\mathbf{w_0}, \mathbf{p_0})$.

Output: $(\mathbf{w}_i, \mathbf{p}_i)_{i=0}^{N}$

for $i \to 1$ **to** N **do**
  $(\mathbf{w}, \mathbf{p}) \leftarrow (\mathbf{w_{i-1}}, \mathbf{p_{i-1}})$.
  $u \sim \mathcal{N}(0, \mathbf{M})$
  $\bar{\mathbf{p}} \leftarrow \rho\mathbf{p} + \sqrt{1-\rho^2}u$
  $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \Phi_{\epsilon,H}^{L}(\mathbf{w}, \bar{\mathbf{p}}, \mathbf{G})$ in equation (2.22)
  $\zeta = \min\left[1, \exp(-\delta \mathrm{H})\right]$, $u \sim \mathrm{Unif}(0, 1)$
  if $\zeta > u$ **then**
    $(\mathbf{w}_i, \mathbf{p}_i, \mathbf{G}) \leftarrow (\hat{\mathbf{w}}, \hat{\mathbf{p}}, \mathbf{G})$
  else
    $(\mathbf{w}_i, \mathbf{p}_i, \mathbf{G}) \leftarrow (\mathbf{w}, -\mathbf{p}, -\mathbf{G})$
  end if
end for

A key consideration when one implements partial momentum refreshment is what technique to use to update the momenta - that is, what should the momenta update equation be? Horowitz [74] mixes the momenta with a Gaussian random number with some user-specified mixing angle. In contrast, Akhmatskaya *et al.* [4, 140] and Heide *et al.* [68] introduce a momentum refreshment parameter that controls the extent of the momentum refreshment between observations, with the new momentum being generated from a Gaussian distribution. This thesis utilises the approach outlined in [153, 68, 140]. In this approach, the updated momentum $\bar{\mathbf{p}}$ to generate the next state is then taken to

be:

$$\bar{\mathbf{p}} = \rho\mathbf{p} + \sqrt{1-\rho^2}u \tag{5.1}$$

with probability one, where ρ is the partial momentum refreshment parameter, u is Gaussian with mean zero and covariance = $\mathbf{M}$, and $\mathbf{p}$ is the momenta for the current state [140].

Algorithm 11 Separable Shadow Hamiltonian Hybrid Monte Carlo with Partial Momentum Refreshment Algorithm

Input: L, ϵ, ρ, N and $(\mathbf{w_0}, \mathbf{p_0})$.

Output: $(\mathbf{w_i}, \mathbf{p_i}, b_i)_{i=0}^{N}$

for $i \to 1$ **to** N **do**
  $(\mathbf{w}, \mathbf{p}) \leftarrow (\mathbf{w_{i-1}}, \mathbf{p_{i-1}})$.
  $u \sim \mathcal{N}(0, \mathbf{M})$
  $\bar{\mathbf{p}} \leftarrow \rho p + \sqrt{1-\rho^2}u$
  Apply the pre-processing mapping in equation (2.33) :
  $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \bar{\mathbf{p}})$
  $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \Phi^{L}_{\epsilon,\tilde{H}}(\hat{\mathbf{w}}, \hat{\mathbf{p}})$ in equation (2.19)
  Apply the post-processing mapping in equation (2.34):
  $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}^{-1}(\hat{\mathbf{w}}, \hat{\mathbf{p}})$
  $\zeta = \min\left[1, \exp(-\Delta\tilde{\mathrm{H}})\right]$, $u \sim \mathrm{Unif}(0,1)$
  if $\zeta > u$ **then**
    $(\mathbf{w}_i, \mathbf{p}_i) \leftarrow (\hat{\mathbf{w}}, \hat{\mathbf{p}})$
  else
    $(\mathbf{w}_i, \mathbf{p}_i) \leftarrow (\mathbf{w}, -\mathbf{p})$
  end if
  $b_i = \exp\left(\tilde{\mathrm{H}}(\mathbf{w_i}, \mathbf{p_i}) - \mathrm{H}(\mathbf{w_i}, \mathbf{p_i})\right)$
end for

Note that $\bar{\mathbf{p}}$ has a Gaussian distribution and $\rho \in (0, 1)$. The two extremes that ρ can assume are: when $\rho = 1$, the momentum is never updated and thus possibly affecting the ergodicity [153] of the chain and when $\rho = 0$, the momentum is always updated, which is the behaviour of the original HMC based samplers. The resultant chain preserves some

dynamics, with the exact amount depending on the value of ρ, between the generated samples [68, 4, 140, 74]. The parameter ρ introduces an extra degree of freedom into the sampler and needs to be set [4, 68]. In Section 5.3.2, we assess the sensitivity of the results on the chosen value of ρ. A key aspect to note is that the proposed algorithms will improve the performance without high additional computational cost and typically do not alter the acceptance rates of the original algorithms in practice [140].

The algorithmic description of the PMHMC and PS2HMC algorithms is presented in Algorithms 10 and 11 respectively. Theorem 5.2.1 guarantees that the proposed methods satisfy detailed balance and hence converge to the correct target density.

Theorem 5.2.1. *PMHMC and PS2HMC satisfy detailed balance.*

Proof. The guarantee that the proposed algorithms satisfy detailed balance provided by the use of the same argument in Horowitz [75] which is: after the MH step, the sign of the momentum (and magnetic field for PMHMC) must be reversed in case of rejection. Given that $\bar{\mathbf{p}}$ in (5.1) is Gaussian, the proof that the proposed methods satisfy detailed balance follows that of MHMC and S2HMC respectively. □

Note that the algorithm for PHMC used in this thesis is the same as the PMHMC method outlined in Algorithm 10, except that we use Hamiltonian dynamics in step 6 instead of non-canonical Hamiltonian dynamics corresponding to MHMC.

5.3 Experiment Description

In this section, we outline the settings used for the experiments, the performance metrics used, and we also present the sensitivity analysis for the partial momentum refreshment parameter α in PMHMC, PHMC and PS2HMC.

5.3.1 Experiment settings

In all our experiments, we compare the performance of PMHMC to HMC, MHMC, S2HMC and PHMC using the acceptance rate of the generated samples, ESS, ESS normalised by execution time and the $\hat{R}$ metric for convergence analysis of the MCMC

chains. The target densities considered are multivariate Gaussian distributions with $D \in \{10, 50\}$, JDPs calibrated to financial market data and real-world datasets modelled using BLR. For all the target posteriors considered in this chapter, the momentum refreshment parameter ρ is set to 0.7. This setting worked well on all the targets. Further experiments of the sensitivity to ρ are presented in Section 5.3.2.

The trajectory length and step sizes parameters for the MCMC methods considered in this chapter were set to the values outlined in Table 3.8. The PMHMC method used the same step size as MHMC, while the S2HMC and PS2HMC methods used the same step size as HMC.

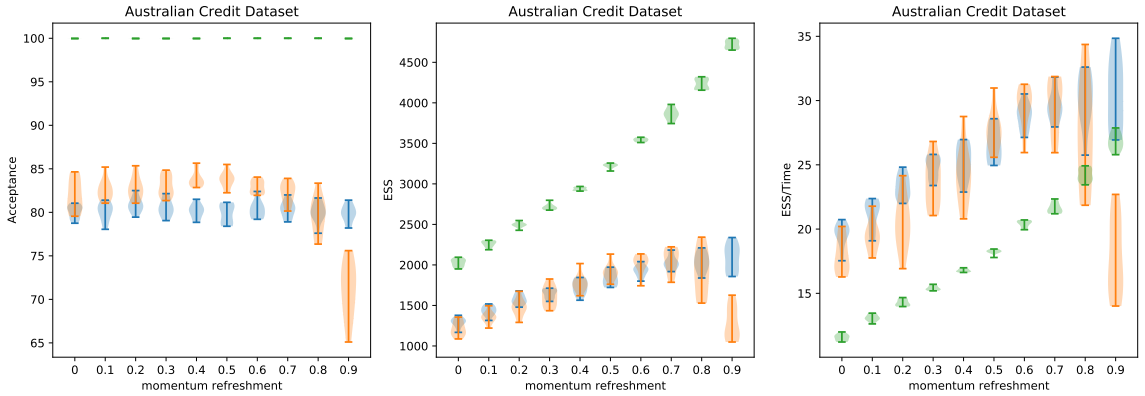
Ten independent chains were run for each method on each target distribution. Three thousand samples were generated for each target, with the first one-thousand samples discarded as burn-in. These settings were sufficient for all the algorithms to converge across all the targets.

5.3.2 Sensitivity to momentum refreshment parameter

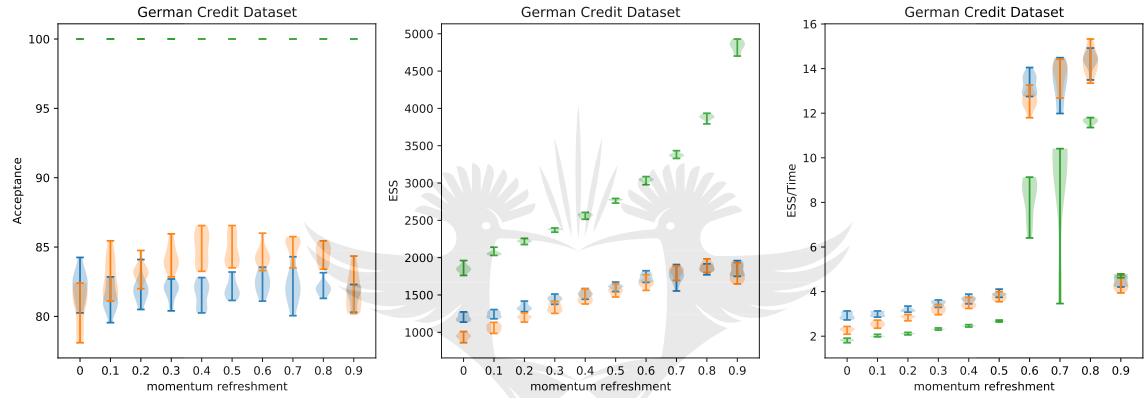
To investigate the effects of varying the momentum refreshment parameter ρ, we ran ten independent chains of PHMC, PMHMC and PS2HMC on two targets for $\rho \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Note that we did not consider the case where $\rho = 1$ as for this case the momentum variable is not refreshed, and this could possibly affect the ergodicity of the proposed methods.

For this analysis, each chain used the same parameters as those in Section 5.3.1. The results are displayed in Figure 5.1. The results show that PHMC, PMHMC and PS2HMC have stable acceptance rates across the different values of ρ on all the targets - indicating that ρ does not impact the acceptance rates of the proposed methods. This result is consistent with the observation of [140]. Furthermore, PS2HMC has the highest acceptance rate, and given that it uses the same step size as PHMC, it highlights the benefits that can be obtained from sampling from the shadow Hamiltonian instead of the true Hamiltonian.

The methods show a general trend of increasing ESS and normalised ESS with increasing ρ for both datasets, with a deterioration of performance occurring at $\rho = 0.9$ as the chain becomes "less ergodic". In addition, PS2HMC has higher ESSs than PHMC



(a)



(b)

Figure 5.1: Acceptance rates, ESS and ESS/Time for ten runs of PHMC (blue), PMHMC (orange), and PS2HMC (green) on the a) Australian and b) German credit datasets with varying choices of the momentum refreshment parameter ρ. The results indicate that the ESS metrics are increasing functions of ρ, while the acceptance rate remains constant across all values of ρ.

and PMHMC for these two targets, but has the lowest time-normalised ESS due to its large execution time. On the German credit dataset, the time-normalised ESS of all the methods jump to a higher level for $\rho > 0.5$, suggesting that the execution time improves for all the methods for $\rho > 0.5$. This phenomenon is not observed on the Australian credit dataset (which has $D = 14$) and is only present on the German credit dataset (which has $D = 25$). This seems to suggest that the impact of partial momentum re-

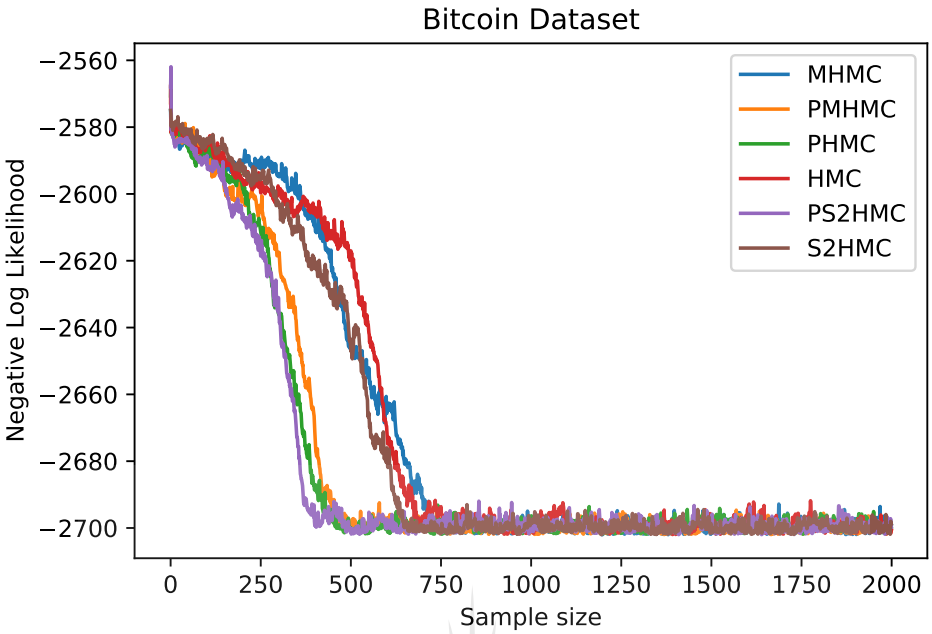


Figure 5.2: Diagnostic trace-plot of the negative log-likelihood showing the convergence behaviour of the algorithms on the Bitcoin dataset. The methods which employ partial momentum refreshment converge faster than the original algorithms. This plot was produced by using $\rho = 0.7$. Two thousand samples were generated with no burn-in period. The other settings are as outlined in Section 5.3.1. The results are for a single run of each algorithm.

freshment becomes more pronounced on a time-normalised ESS as D increases. We intend to investigate this behaviour further in future work.

The automatic tuning of ρ is still an open research problem. We plan to address the automatic tuning of this parameter in future work. As a guideline, higher values of ρ seem to be associated with higher ESS.

5.4 Results and Discussion

We first assess the convergence and mixing behaviour of the algorithms through plotting the convergence profile of the negative log-likelihood in Figure 5.2. This plot is from the point at which the first sample is generated until the last sample is generated. The results indicate that the methods which employ partial momentum refreshment converge

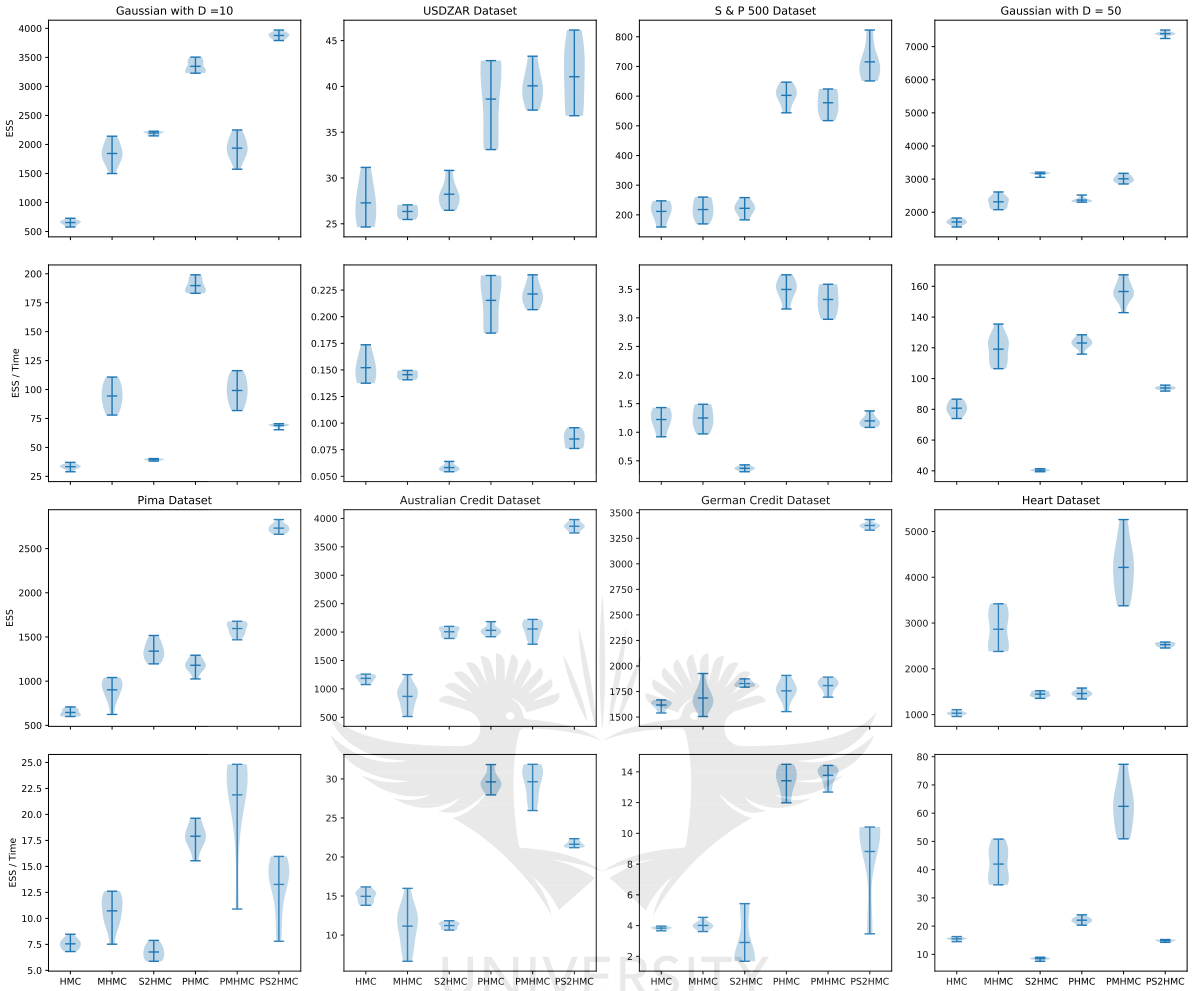


Figure 5.3: Results for the targets over ten runs of each method. For each dataset, the plots on the first row show the multivariate effective sample size and the plots on the second row show the multivariate effective sample size normalised by execution time (in seconds). The dark horizontal line in each violin plot represents the mean value over ten runs of each algorithm. For all the plots, the larger the value, the better the method.

faster than the original algorithms. This shows that the mixing speed of Hamiltonian dynamics-based samplers can be improved by including partial momentum refreshment.

The performance of the algorithms across different metrics is shown in Figure 5.3 and Tables 5.1 to 5.3. In Figure 5.3, the plots on the first row for each dataset show the ESS, and the plots on the second row show the ESS normalised by execution time. The

Table 5.1: JDP results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. AR stands for the acceptance rate of the generated samples post the burn-in period.

USDZAR dataset						
Metric	HMC	MHMC	S2HMC	PHMC	PMHMC	PS2HMC
AR	**99.99**	99.93	99.97	**99.99**	99.97	**99.99**
ESS	27	26	28	38	40	**41**
t	**179**	181	484	181	181	482
ESS/t	0.15	0.14	0.05	0.21	**0.22**	0.08
$\hat{R}$ max	1.03	1.02	1.03	1.01	1.02	**1.00**
S&P 500 dataset						
AR	92.72	92.68	**99.3**	92.33	92.58	**99.3**
ESS	211	217	222	602	577	**715**
t	173	174	601	**172**	173	597
ESS/t	1.22	1.24	0.36	**3.49**	3.32	1.19
$\hat{R}$ max	1.03	1.02	1.04	**1.01**	1.02	**1.01**

results are for ten runs of each algorithm. The execution time t in Figure 5.3 and Tables 5.1 to 5.3 is in seconds. The results in Tables 5.1 to 5.3 are the mean results over ten runs for each algorithm. We use the mean values over the ten runs in Tables 5.1 to 5.3 to form our conclusions about the performance of the algorithms.

The results in Figure 5.3 and Tables 5.1 to 5.3 show that, as expected, MHMC outperforms HMC on all but one target being the Australian credit dataset. This is in line with the behaviour which we saw in Section 4.3 of Chapter 4. Furthermore, these results are in line with what has been previously observed in the literature [156, 26].

We also notice that S2HMC produces higher acceptance rates than HMC across all the targets while using the same step size. This shows the benefits that can be derived from sampling from shadow Hamiltonians. We further find that the methods that partially refresh the momentum produce similar acceptance rates to the original algorithms. In addition, the results show that the PS2HMC and S2HMC algorithms

Table 5.2: Multivariate Gaussian distribution results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. AR stands for the acceptance rate of the generated samples post the burn-in period.

Gaussian with D = 10						
Metric	HMC	MHMC	S2HMC	PHMC	PMHMC	PS2HMC
AR	86.21	80.96	**100.00**	86.22	82.86	**100.00**
ESS	653	1 844	2 197	3 345	1 936	**3 878**
t	19.61	**19.52**	55.67	17.63	**19.52**	56.25
ESS/t	33.33	94.45	39.48	**189.78**	99.17	68.95
$\hat{R}$ max	1.03	1.02	1.02	1.03	1.02	**1.01**
Gaussian with D = 50						
AR	73.50	83.32	99.68	73.56	82.32	**99.92**
ESS	1 705	2 313	3 164	2 366	3 008	**7 390**
t	21.12	19.42	78.28	19.22	**19.20**	78.73
ESS/t	80.71	119.11	40.43	123.14	**156.61**	93.86
$\hat{R}$ max	**1.01**	1.03	1.02	1.03	1.02	**1.01**

consistently produce high acceptance rates across all the targets and outperform the other methods on this metric.

The results also indicate that PMHMC and PS2HMC outperform MHMC and S2HMC across all the targets considered on the ESS and ESS normalised by execution time metrics, respectively. Furthermore, PHMC outperforms HMC - which was also observed by Horowitz [74]. This result was also observed in the context of RMHMC in [68]. This shows the significant benefit partial momentum updates can provide to the sampling properties of HMC based samplers. However, the real performance gains are realised when the momentum refreshment parameter ρ has been appropriately tuned.

In general, we find that the algorithms with partial momentum refreshment have similar execution times compared to the original algorithms, except on the German credit dataset, where the execution time of the partial momentum refreshment algorithms

Table 5.3: BLR results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. AR stands for the acceptance rate of the generated samples post the burn-in period.

Pima dataset						
Metric	HMC	MHMC	S2HMC	PHMC	PMHMC	PS2HMC
AR	78.195	82.35	99.89	78.39	80.60	**99.94**
ESS	645	902	1 339	1 179	1 595	**2 731**
t	85.54	84.13	199	**65**	76	215
ESS/t	7.55	10.72	6.76	17.90	**21.88**	13.26
$\hat{R}$ max	**1.00**	1.02	**1.00**	**1.00**	1.01	**1.00**
Heart dataset						
AR	81.69	79.10	99.96	81.93	83.67	**100.00**
ESS	1 030	2 864	1 447	1 460	**4 215**	2 528
t	**66**	68	170	**66**	67	170
ESS/t	15.48	41.96	8.48	22.09	**62.42**	14.83
$\hat{R}$ max	1.02	**1.01**	1.02	**1.01**	**1.01**	1.02
Australian credit dataset						
AR	79.88	82.06	**100**	80.33	82.22	**100**
ESS	1 184	867	2 005	2 030	2 053	**3 862**
t	79.18	77.82	178	**68**	69	178
ESS/t	14.96	11.14	11.21	29.61	**29.63**	21.62
$\hat{R}$ max	1.01	1.04	**1.00**	**1.00**	**1.00**	**1.00**
German credit dataset						
AR	81.89	81.41	**100.0**	82.03	84.9	**100.0**
ESS	1 617	1 686	1 828	1 756	1 808	**3 375**
t	421.79	420.51	630.08	**130.01**	131.11	382.62
ESS/t	3.83	4.01	2.90	13.42	**13.77**	8.82
$\hat{R}$ max	1.02	**1.01**	1.02	**1.01**	**1.01**	**1.01**

drops. This is due to the phenomenon that we outlined in Section 5.3.2. This particular aspect of the method requires further investigation, which we plan to conduct in future work. It is worth highlighting that the S2HMC and PS2HMC algorithms have the most considerable execution times across all the targets. This is mainly due to the multiple times that the shadow Hamiltonian is evaluated. This affects the time-normalised ESS performance of these algorithms.

We find that PS2HMC outperforms all the methods on an ESS basis on all the targets, except on the Heart dataset where PMHMC outperforms all the methods. PMHMC outperforms all the methods on a time-normalised ESS basis on the BLR datasets, with mixed results on the Gaussian and JDP targets. We further find that S2HMC sometimes outperforms, on an ESS basis, all the other methods except for PS2HMC, for example, on the Gaussian targets and the German credit dataset. This indicates that S2HMC sometimes precludes, baring the slow execution time, the need for PMHMC in practice.

It is worth noting that all the methods produce low ESSs on the JDP datasets, indicating the difficulty of sampling from JDPs. The results of the $\hat{R}$ metric across all the targets show that the methods have converged, with the algorithms displaying similar $\hat{R}$ metrics without an outright outperformer on all the targets.

5.5 Conclusion

In this chapter, we introduced partial momentum refreshment to MHMC and S2HMC. We showed that these modifications result in improved sampling performance over MHMC and S2HMC with minimal additional computational costs. We find that PS2HMC outperforms all the methods on an ESS basis on the majority of the targets, with PMHMC outperforming on the BLR datasets on a time-normalised ESS basis. These results indicate the overall efficacy of incorporating partial momentum refreshment in Hamiltonian dynamics-based samplers.

A limitation of the proposed methods is the need to tune the momentum refreshment parameter. Although typically larger parameter values improve the effective sample sizes, a more robust approach to selecting the parameter is still required. This work can be improved by establishing a heuristic or an automated approach to tune the momentum

refreshment parameter. Furthermore, exploring techniques for accelerating the execution time of S2HMC and PS2HMC is worthwhile as these two algorithms produce the lowest execution times, which hampers their performance on a normalised ESS basis.

In the next chapter, we extend MHMC by deriving the fourth-order shadow Hamiltonian associated with the integration scheme used in MHMC. We then use this modified Hamiltonian to construct the new SMHMC sampler.

Chapter 6

Shadow Magnetic Hamiltonian Monte Carlo

6.1 Introduction

In the preceding chapter, we showed that PMHMC improves the sampling performance of MHMC through the incorporation of partial momentum refreshment. Sampling from an integrator-dependent shadow or modified target density has been utilised to boost the acceptance rates of HMC. This leads to more efficient sampling as the shadow Hamiltonian is better conserved by the integrator than the true Hamiltonian [77]. Sampling from the modified Hamiltonian associated with the numerical integrator used in MHMC is yet to be explored in the literature. This chapter aims to address this gap in the literature by combining the benefits of the non-canonical Hamiltonian dynamics of MHMC with those achieved by targeting the modified Hamiltonian. We first determine the modified Hamiltonian associated with the MHMC integrator and use this to construct a novel method, which we refer to as SMHMC, that leads to better sampling behaviour when compared to MHMC while leaving the target distribution invariant. Furthermore, the SMHMC algorithm employs partial momentum refreshment for the momenta generation. The new SMHMC method is compared to MHMC and PMHMC. The material in this chapter is currently under review at an international journal.

6.2 Background

It has been previously confirmed that the performance of HMC suffers from the deterioration in acceptance rates due to numerical integration errors that arise as a result of large integration step sizes ϵ, or as the system size increases [77, 68]. As MHMC is an extension of HMC, and becomes HMC when the magnetic component is absent, one would expect that it also suffers from the pathology of a rapid decrease in acceptance rates as the system size increases. The results in Figure 6.1 suggest that MHMC may indeed suffer from a deterioration in acceptance rates.

The rapid decrease in acceptance rates results in significant auto-correlations between the generated samples, thus requiring large sample sizes. The deterioration of the acceptance rates can be reduced by using more accurate higher-order integrators, by using smaller step sizes, or by employing shadow Hamiltonians [140]. The first two approaches tend to be more computationally expensive than the latter approach [68, 140]. In this chapter, we explore the approach of utilising shadow Hamiltonians to enhance the sampling performance of MHMC.

Samplers based on shadow Hamiltonians have been successfully employed to address the deterioration of sample acceptance as the system, and step sizes increase and also lead to more efficient sampling of the target posterior [4, 155, 77]. The shadow Hamiltonians are constructed by performing backward error analysis of the integrator and, as a result, are better conserved when compared to the true Hamiltonian [63]. Numerous approaches have been proposed for sampling from shadow Hamiltonians of various numerical integrators [4, 68, 155, 77]. Sampling from the modified Hamiltonian associated with the numerical integrator employed in MHMC is yet to be explored in the literature. In this chapter, we address this gap in the literature by deriving the fourth-order modified Hamiltonian associated with the numerical integrator in MHMC. From this modified Hamiltonian, we create the new SMHMC method, which satisfied detailed balance. We compare the performance of the proposed method to MHMC and PMHMC across the Banana shaped distribution, multivariate Gaussian distribution, Protein dataset modeled using BNNs, and the Heart dataset modeled using BLR as highlighted in Table 3.2.

We proceed in this chapter by first deriving the shadow Hamiltonian corresponding

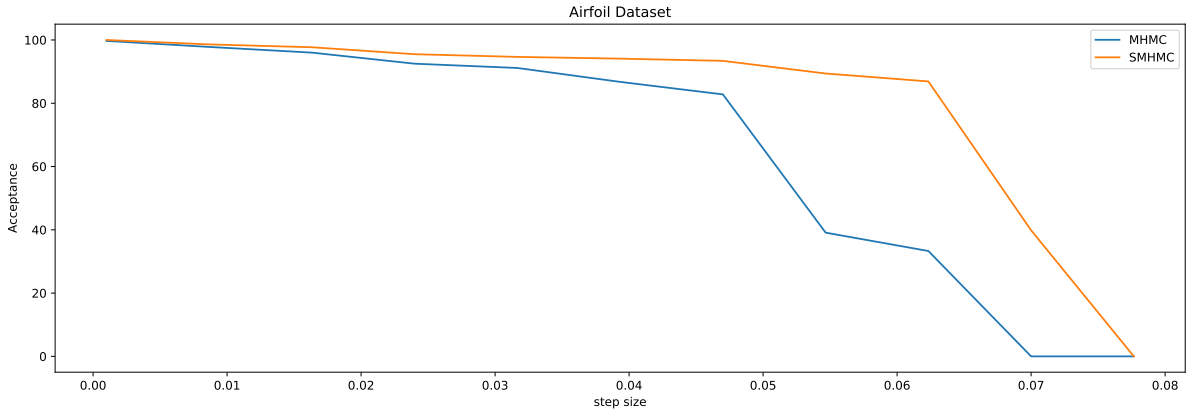


Figure 6.1: Impact of the step size on the acceptance rate of SMHMC and MHMC on the Airfoil dataset. The results show that SMHMC maintains higher acceptance rates than MHMC as the step size increases. Three thousand samples were generated for each run, with the first one thousand samples being the burn-in period. The results displayed in this plot were averaged over five runs of the algorithms.

to the leapfrog-like integration scheme in MHMC, after which we utilise this shadow Hamiltonian to construct the novel SMHMC algorithm.

6.3 Shadow Hamiltonian for MHMC

In this chapter, we focus on a fourth-order truncation of the shadow Hamiltonian under the leapfrog-like integrator in equation (2.22). Since the leapfrog-like integrator is second-order accurate ($\mathcal{O}^2$) [156], the fourth-order truncation is conserved with higher accuracy ($\mathcal{O}^4$) by the intergrator than the true Hamiltonian. In Theorem 6.3.1, we derive the fourth-order shadow Hamiltonian under the leapfrog-like integrator.

Theorem 6.3.1. *Let $H : R^d \times R^d = R$ be a smooth Hamiltonian function. The fourth–order shadow Hamiltonian function $\hat{\mathrm{H}} : R^d \times R^d = R$ corresponding to the leapfrog-like integrator used in MHMC is given by:*

$$\hat{\mathrm{H}}(\mathbf{w}, \mathbf{p}) = \mathrm{H}(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12}\left[K_{\mathbf{p}} U_{\mathbf{ww}} K_{\mathbf{p}} + K_{\mathbf{p}} \mathbf{G} K_{\mathbf{pp}} U_{\mathbf{w}}\right] - \frac{\epsilon^2}{24}\left[U_{\mathbf{w}} K_{\mathbf{pp}} U_{\mathbf{w}}\right] + \mathcal{O}(\epsilon^4) \quad (6.1)$$

Proof. As outlined in Tripuraneni *et al.* [156], the Hamiltonian vector field $\overrightarrow{H} =$

$\nabla_{\mathbf{p}} H \nabla_{\mathbf{w}} + (-\nabla_{\mathbf{w}} + \mathbf{G}\nabla_{\mathbf{p}} H)\nabla_{\mathbf{p}} = \overrightarrow{A} + \overrightarrow{B}$ will generate the exact flow corresponding to exactly simulating the MHMC dynamics [156]. We obtain the shadow Hamiltonian via the separability of the true Hamiltonian [156]. The leapfrog-like integration scheme in equation (2.22) splits the Hamiltonian as: $H(\mathbf{w}, \mathbf{p}) = H_1(\mathbf{w}) + H_2(\mathbf{p}) + H_1(\mathbf{w})$ and exactly integrates each sub-Hamiltonian [156]. Using the BCH [64] formula we obtain:

$$\begin{aligned} \Phi_{\epsilon,H}^{frog} &= \Phi_{\epsilon,H_1(\mathbf{w})} \circ \Phi_{\epsilon,H_2(\mathbf{p})} \circ \Phi_{\epsilon,H_1(\mathbf{w})} \\ &= \exp\left(\frac{\epsilon}{2}\overrightarrow{B}\right) \circ \exp\left(\epsilon\overrightarrow{A}\right) \circ \exp\left(\frac{\epsilon}{2}\overrightarrow{B}\right) \\ &= \mathrm{H}(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12}\{K, \{K, U\}\} - \frac{\epsilon^2}{24}\{U, \{U, K\}\} + \mathcal{O}(\epsilon^4) \end{aligned} \tag{6.2}$$

where the non-canonical Poisson brackets [32, 26] are defined as:

$$\begin{aligned} \{f, g\} &= [\nabla_{\mathbf{w}} f, \nabla_{\mathbf{p}} f] \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{G} \end{bmatrix} [\nabla_{\mathbf{w}} g, \nabla_{\mathbf{p}} g]^T \\ &= -\nabla_{\mathbf{p}} f \nabla_{\mathbf{w}} g + \nabla_{\mathbf{w}} f \nabla_{\mathbf{p}} g + \nabla_{\mathbf{p}} f \mathbf{G} \nabla_{\mathbf{p}} g \end{aligned} \tag{6.3}$$

and collapse to the canonical Poisson brackets when $\mathbf{G} = \mathbf{0}$ [32, 26]. The corresponding derivatives from the non-canonical Poisson brackets are presented in Appendix D. The shadow Hamiltonian for the leapfrog-like integrator is then:

$$\hat{\mathrm{H}}(\mathbf{w}, \mathbf{p}) = \mathrm{H}(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12}\left[K_{\mathbf{p}} U_{\mathbf{ww}} K_{\mathbf{p}} + \underbrace{K_{\mathbf{p}} \mathbf{G} K_{\mathbf{pp}} U_{\mathbf{w}}}_{A}\right] - \frac{\epsilon^2}{24}\left[U_{\mathbf{w}} K_{\mathbf{pp}} U_{\mathbf{w}}\right] + \mathcal{O}(\epsilon^4) \tag{6.4}$$

where A is the factor induced by the presence of the magnetic field $\mathbf{G}$. When $\mathbf{G} = \mathbf{0}$, the shadow in (6.4) becomes the shadow for Hamiltonian dynamics [77, 155] in equation (2.30). It is worth noting that the shadow Hamiltonian in (6.4) is conserved to fourth-order [68, 155, 140], and is thus more accurately conserved by leapfrog-like integrator than the true Hamiltonian [151]. □

6.4 Proposed Shadow Magnetic Algorithm

We now present the SMHMC algorithm, which combines non-canonical Hamiltonian dynamics in MHMC with the high conservation property of shadow Hamiltonians. The

benefits of employing non-canonical Hamiltonian dynamics in MHMC have already been established in [156, 59, 116], while the advantages of shadow Hamiltonians in general are presented in [155, 68, 99, 112, 4]. We combine these two concepts to create a new sampler that outperforms MHMC across various performance metrics.

An analysis of the shadow Hamiltonian corresponding to MHMC in equation (6.1) shows that the conditional density for the momenta $\pi_H(\mathbf{p}|\mathbf{w})$ is not Gaussian. This suggests that if we fully re-sample the momenta from a normal distribution, as we did in Chapter 5, we will attain a sampler that does not satisfy detailed balance [68, 4]. This necessitates computationally intensive momentum generation [77] or partial momentum refreshment [140, 4] with an MH step. In this chapter, we utilise the partial momentum refreshment procedure outlined in [4, 68], in which a Gaussian noise vector $u \sim \mathcal{N}(0, \mathbf{M})$ is drawn. The momentum proposal is then produced via the mapping:

$$R(\mathbf{p}, u) = \left(\rho\mathbf{p} + \sqrt{1-\rho^2}u, -\sqrt{1-\rho^2}\mathbf{p} + \rho u\right) \tag{6.5}$$

The new parameter, which we refer to as the momentum refreshment parameter, $\rho = \rho(\mathbf{w}, \mathbf{p}, u)$ takes values between zero and one, and controls the extent of the momentum retention [116, 68, 140]. When ρ is equal to one, the momentum is never updated and when ρ is equal to zero, the momentum is always updated [116]. The momentum proposals are then accepted according to the modified non-separable shadow Hamiltonian given as $\bar{\mathrm{H}}(\mathbf{w}, \mathbf{p}, u) = \hat{\mathrm{H}}(\mathbf{w}, \mathbf{p}) + \frac{1}{2}u\mathbf{M}^{-1}u$. The updated momentum is then taken to be $\rho\mathbf{p} + \sqrt{1-\rho^2}u$ with probability:

$$\omega := \max\{1, \exp(\bar{\mathrm{H}}(\mathbf{w}, \mathbf{p}, u) - \bar{\mathrm{H}}(\mathbf{w}, R(\mathbf{p}, u)))\}. \tag{6.6}$$

The incomplete refreshment of the momentum produces a chain which saves some of the behaviour between neighbourhood samples [116, 74, 68, 4, 140]. In Section 6.5.2, we assess the sensitivity of the sampling results on the user-specified value of ρ. An algorithmic description of the SMHMC sampler is provided in Algorithm 12. It is worth noting from Algorithm 12 that the SMHMC sampler uses two reversible MH steps, which implies that the resulting Markov chain is no longer reversible [68, 4, 140]. By breaking the detailed balance condition, it is no longer immediately clear that the target density is stationary, and so this must be demonstrated [68, 140].

Theorem 6.4.1. *The SMHMC algorithm leaves the importance target distribution invariant.*

Proof. The proof of theorem 6.4.1 is obtained in Appendix A of the paper by Radivojevic and Akhmatskay [140]. The proof involves showing that the addition of step 4 in Algorithm 12 leaves the target invariant. The result follows from [140] by making use of the fact that the explicit form of the shadow Hamiltonian, which has the additional magnetic component $A = K_{\mathbf{p}}\mathbf{G}K_{\mathbf{pp}}U_{\mathbf{w}}$ in equation (6.4) in our case, is not required for the proof [140, 68]. □

Algorithm 12 Shadow Magnetic Hamiltonian Monte Carlo Algorithm

Input: L, ϵ, ρ, N, $\mathbf{G}$, $(\mathbf{w_0}, \mathbf{p_0})$.

Output: $(w_i, p_i, b_i)_{i=0}^{N}$

for $i \to 1$ **to** N **do**
  $(\mathbf{w}, \mathbf{p}) \leftarrow (\mathbf{w_{i-1}}, \mathbf{p_{i-1}})$.
  $u \sim \mathcal{N}(0, \mathbf{M})$
  $\bar{p} \leftarrow \rho p + \sqrt{1-\rho^2}u$ with probability ω in equation (6.6).
  $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \Phi_{\epsilon,H}^{L}(\mathbf{w}, \bar{\mathbf{p}}, \mathbf{G})$ in equation (2.22)
  $\zeta = \min\left[1, \exp(-\delta\hat{\mathrm{H}})\right]$, $u \sim \mathrm{Unif}(0,1)$
  if $\zeta > u$ **then**
    $(\mathbf{w}_i, \mathbf{p}_i, \mathbf{G}) \leftarrow (\hat{\mathbf{w}}, \hat{\mathbf{p}}, \mathbf{G})$
  else
    $(\mathbf{w}_i, \mathbf{p}_i, \mathbf{G}) \leftarrow (\mathbf{w}, -\mathbf{p}, -\mathbf{G})$
  end if
  $b_i = \exp\left(\hat{\mathrm{H}}(\mathbf{w_i}, \mathbf{p_i}) - \mathrm{H}(\mathbf{w_i}, \mathbf{p_i})\right)$
end for

6.5 Experiment Description

In this section, we outline the settings used for the experiments, the performance metrics used, and we also present the sensitivity analysis for the partial momentum refreshment parameter ρ in SMHMC and PMHMC.

6.5.1 Experiment settings

In our analysis, we compare the performance of SMHMC against MHMC and PMHMC across the Banana shaped distribution, multivariate Gaussian distribution with $D = 10$, the Protein dataset modeled using a BNN and the Heart dataset modeled using BLR. We assess the performance using the acceptance rate, ESS, time-normalised ESS metrics and assess the convergence behaviour of the chains using the $\hat{R}$ metric. Note that for all our experiments in this chapter, we set $\mathbf{M} = \mathbf{I}$ which is the common approach in practice [17].

For all the target posteriors used in this chapter, the momentum refreshment parameter ρ is set to 0.7, which is consistent with the settings used in Chapter 5. This setting worked well on all the targets. Further experiments of the sensitivity to ρ are presented in Section 6.5.2. The trajectory length and step size parameters for the MCMC methods considered in this chapter were set to the values outlined in Table 3.8. Note that the SMHMC and PMHMC methods used the same step size as MHMC.

Ten independent chains were run for each method on each target distribution. Three thousand samples were generated for each target, with the first one-thousand samples discarded as burn-in. These settings were sufficient for all the algorithms to converge on all the targets.

6.5.2 Sensitivity to momentum refreshment parameter

We investigate the effects of varying the momentum refreshment parameter ρ on the sampling performance of the proposed shadow Hamiltonian method. Ten chains, starting from different positions, of the PMHMC and shadow MHMC algorithms were ran on the Australian credit dataset for $\rho \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Note that we exclude $\rho = 1.0$ from the analysis due to the same reasons outlined in Section 5.3.2. Figure 6.2 shows the results for the Australian credit dataset. The results show that PMHMC and SMHMC have stable acceptance rates across the different values of ρ on all this target. SMHMC has higher acceptance rates and ESS than PMHMC for the same step size. However, due to the high execution time of SMHMC, PMHMC produced better time-normalised ESS compared to SMHMC. The methods show a general trend of

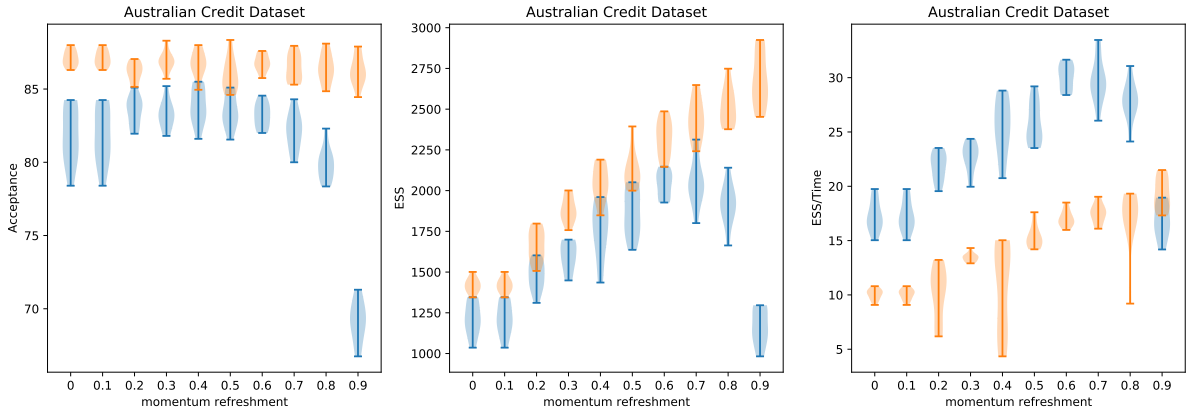


Figure 6.2: Acceptance rates, ESS and ESS/Time for ten chains of PMHMC (blue) and SMHMC (orange) the Australian credit dataset with varying choices of ρ. The ESS metrics are an increasing function of ρ with the acceptance rate of SMHMC being larger than PMHMC for the same step size ϵ.

increasing ESS and time-normalised ESS with increasing ρ, which is consistent with what was observed in Section 5.3.2 for other Hamiltonian methods that incorporate partial momentum refreshment.

6.6 Results and Discussion

Figure 6.3 shows the diagnostic trace-plots of the negative log-likelihood across various target posteriors. The results show that all three methods have converged on the four target densities analysed in this chapter.

Table 6.1: Multivariate Gaussian distribution with $D = 10$ results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. AR stands for the acceptance rate of the generated samples post the burn-in period.

Gaussian with $D = 10$			
Metric	MHMC	PMHMC	SMHMC
AR	80.96	82.86	**84.64**
ESS	1 844	1 936	**2 546**
t	**19.52**	**19.52**	69.01
ESS/t	94.45	**99.17**	36.90
$\hat{R}$ max	1.02	1.02	**1.01**

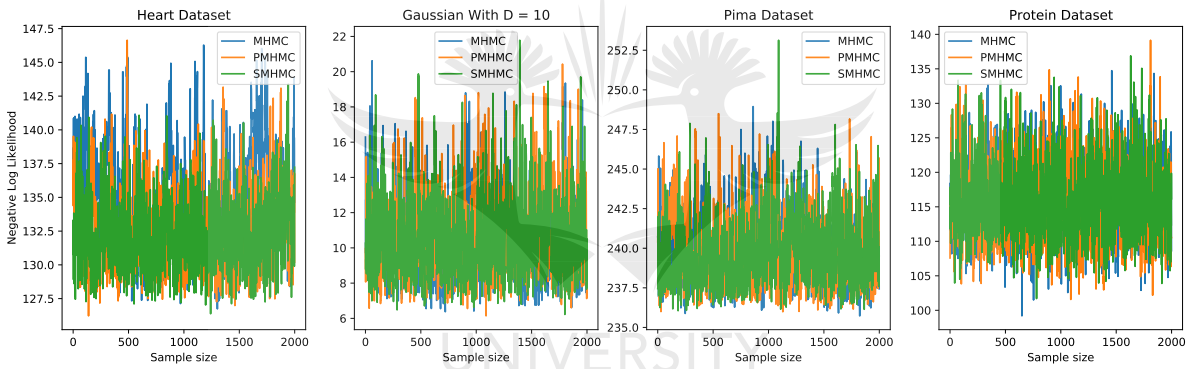


Figure 6.3: Diagnostic trace-plots of the negative log-likelihood across various targets averaged over ten runs of each method. These results show that all the MCMC methods have converged on all the targets.

The performance of the algorithms across different metrics is shown in Figure 6.4 and Tables 6.1 to 6.4. In Figure 6.4, the plots on the first row for each dataset show the effective sample size, and the plots on the second row show the effective sample size normalised by execution time. The results are for the ten runs of each algorithm. The execution time t in Figure 6.4 and Tables 6.1 to 6.4 is in seconds. The results in Tables 6.1 to 6.4 are the mean results over the ten runs for each algorithm. We use the mean values over the ten runs in Tables 6.1 to 6.4 to form our conclusions about the

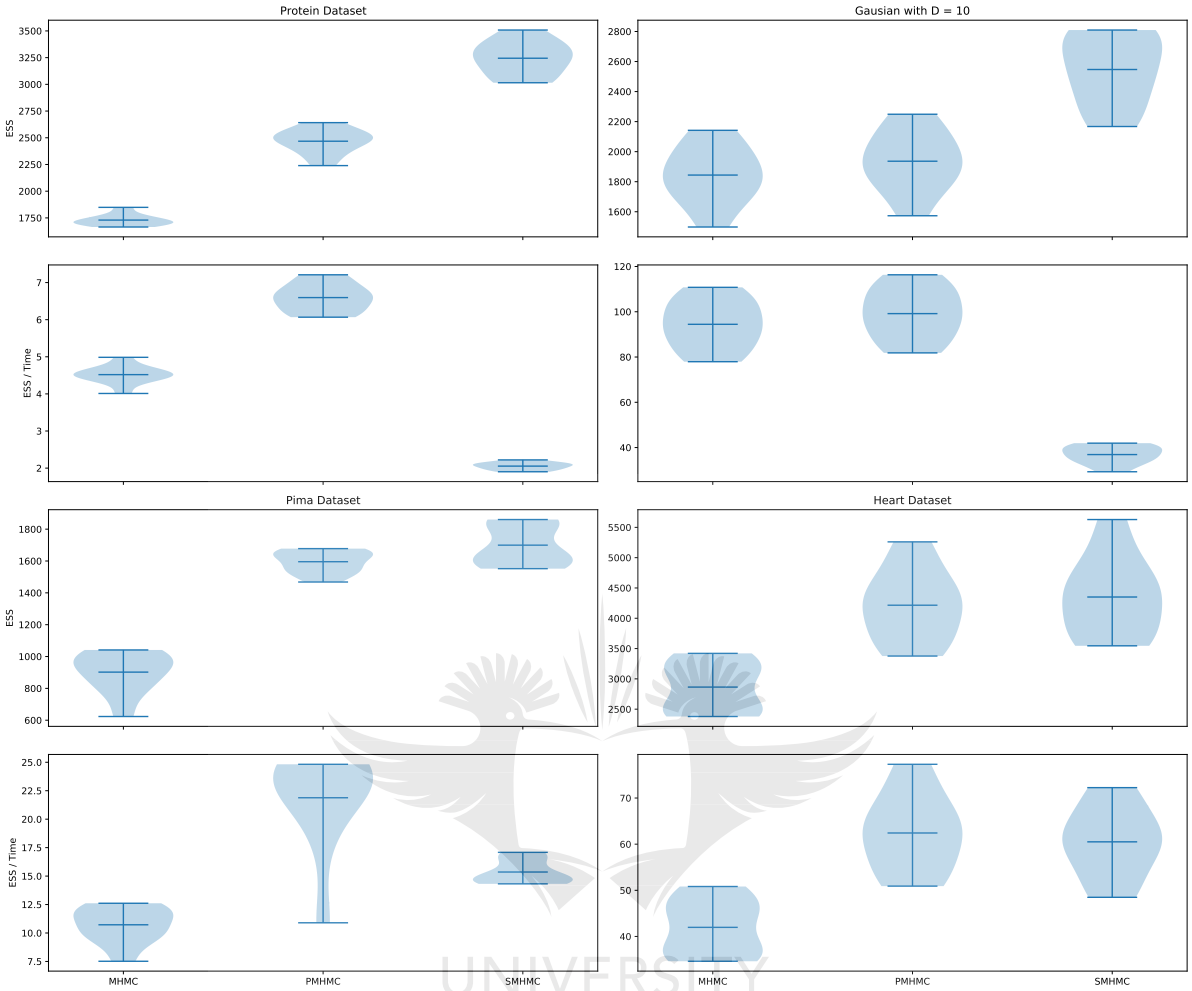


Figure 6.4: Results for the datasets over ten runs of each method. For each dataset, the plots on the first row show the multivariate effective sample size and the plots on the second row show the multivariate effective sample size normalised by execution time (in seconds). For all the plots, the larger the value, the better the method. The dark horizontal line in each violin plot represents the mean value over ten runs of each algorithm.

performance of the algorithms.

Tables 6.1 to 6.4 show that SMHMC produces the highest acceptance rate across all the targets, which is consistent with what we observed for other shadow Hamiltonian algorithms in Chapter 5. Furthermore, the SMHMC algorithm produces the largest ESS on all the targets. In particular, it outperforms PMHMC, which shows that the

Table 6.2: Protein dataset results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. AR stands for the acceptance rate of the generated samples post the burn-in period.

Protein Dataset			
Metric	MHMC	PMHMC	SMHMC
AR	80.28	79.4	**82.09**
ESS	1 729	2 467	**3 244**
t	**373**	374	1 579
ESS/t	4.51	**6.54**	2.05
$\hat{R}$ max	1.01	**1.00**	**1.00**

method is doing something extra than just incorporating partial momentum refreshment into the MHMC. However, the source of the outperformance of SMHMC seems to be mostly driven by the incorporation of the partial momentum refreshment, highlighting the benefits of utilising partial momentum refreshment in Hamiltonian dynamics-based samplers in general.

The results show that the MHMC and PMHMC produce the lowest execution times across all the targets, with SMHMC having the largest execution time, sometimes as much as two times that of the MHMC method. The large execution time of SMHMC can be attributed to the multiple times that the shadow Hamiltonian is evaluated, as well as the extra MH step for the momenta generation. The slow execution time is the key drawback of SMHMC, and hinders the performance of the method on a time-normalised ESS basis. We find that PMHMC outperforms all the methods on a time-normalised ESS basis. SMHMC outperforms MHMC on the BLR datasets on a time-normalised ESS basis, with MHMC outperforming SMHMC on the other targets on the same basis. Furthermore, the $\hat{R}$ metric shows that all the methods have converged, with PMHMC and SMHMC producing marginally better convergence behaviour compared to MHMC.

Table 6.3: Heart dataset results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. AR stands for the acceptance rate of the generated samples post the burn-in period.

Heart Dataset			
Metric	MHMC	PMHMC	SMHMC
AR	79.10	83.67	**84.66**
ESS	2 864	4 215	**4 350**
t	68.24	**67.5**	71.90
ESS/t	41.96	**62.42**	60.50
$\hat{R}$ max	1.01	1.01	**1.00**

Table 6.4: Pima dataset results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. AR stands for the acceptance rate of the generated samples post the burn-in period.

Pima Dataset			
Metric	MHMC	PMHMC	SMHMC
AR	82.35	80.60	**84.80**
ESS	902	1 595	**1 699**
t	84.13	**72.91**	110.66
ESS/t	10.72	**21.88**	15.35
$\hat{R}$ max	1.02	**1.01**	**1.01**

6.7 Conclusion

In this chapter, we introduced the novel SMHMC algorithm, which combines the non-canonical dynamics of MHMC with the benefits of sampling from a shadow Hamiltonian. This combination results in improved exploration of the posterior when compared to MHMC. The empirical results show that the new algorithm provides a significant improvement on the MHMC algorithm in terms of higher acceptance rates and larger effective sample sizes, even as the dimensionality of the problem increases.

A primary limitation of the proposed algorithm is the computational time associated with the method, mainly since it involves the computation of the Hessian matrix of the target distribution. This leads to poor performance on a time-normalised ESS basis. A straightforward approach to circumvent the computational burden is to use closed-form expressions for the first-order derivatives and the Hessian matrix. This approach, however, restricts the possible targets that can be considered. We aim to address this issue in the future by using a surrogate model to approximate the shadow Hamiltonian during the burn-in period of the method as an active learning task.

Another limitation of the method is the need to tune the momentum refreshment parameter. Although typically higher values of the parameter improve the effective sample sizes, a more robust approach to selecting the parameter is still required. In future work, we plan on improving the proposed method by establishing an automated approach to tune the momentum refreshment parameter.

In the next chapter, we address a crucial impediment in the general use of the S2HMC method, which is the tuning the parameters of S2HMC. Tuning of parameters of shadow Hamiltonian methods is yet to be explored in the literature. The next chapter aims to fill this gap and simultaneously make S2HMC more accessible to non-expert users.

Chapter 7

Adaptive Shadow Hamiltonian Monte Carlo

7.1 Introduction

As already highlighted in this thesis, HMC has two main practical issues. The first is the deterioration in acceptance rates as the system size increases, and the second is its sensitivity to two user-specified parameters: the step size and trajectory length. The former issue is addressed by sampling from an integrator-dependent modified or shadow density and compensating for the induced bias via importance sampling. The latter issue is addressed by adaptively setting the HMC parameters, with the state-of-art method being the NUTS algorithm. The automatic tuning of parameters of shadow Hamiltonian methods is yet to be considered in the literature. In this chapter, we combine the benefits of NUTS with those attained by sampling from the shadow density by adaptively setting the trajectory length and step size of S2HMC to make it more accessible to non-expert users. This leads to a new algorithm which we refer to as adaptive S2HMC, that shows improved performance over S2HMC and NUTS across various targets and leaves the target density invariant. The work in this chapter was published in the following journal article:

1. **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Adaptively Setting the Path Length for Separable Shadow Hamiltonian Hybrid Monte Carlo.* IEEE Access, vol.

9, pp. 138598-138607.

7.2 Proposed Adaptive Shadow Algorithm

The performance of HMC suffers from two significant practical issues. The first practical issue relates to the degeneration in acceptance rates due to numerical integration errors as the system size grows [77, 68]. This deterioration in acceptance rates results in significant auto-correlations in the generated samples, necessitating the generation of large sample sizes. The second difficulty with HMC is the need to tune performance-sensitive parameters in the step size and trajectory length. A small trajectory length leads to the display of random walk behaviour [73, 2], and one that is too large leads to wasted computation [73, 2]. In contrast, small step sizes are computationally wasteful, leading to correlated samples and poor mixing. Large step sizes compound discretisation errors, which also lead to lousy mixing [73, 2]. Finding the optimal values for these parameters is of paramount importance but is far from trivial [73, 2].

An approach to address the rapid decrease in acceptance rates with increases in system size is to use modified or shadow Hamiltonian based samplers. These modified Hamiltonian methods leverage backward error analysis of the numerical integrator, which results in higher-order conservation of the shadow Hamiltonian relative to the true Hamiltonian [63]. Numerous methods have been put forward for sampling from a shadow Hamiltonian [4, 68, 155, 77, 99].

Sweet *et al.* [155] present S2HMC which leverages a processed leapfrog integrator that results in a separable shadow Hamiltonian. The separable shadow Hamiltonian reduces the need for computationally intensive momentum generation [77] or partial momentum refreshment [140, 4] evaluations that are necessitated by non-separable shadow Hamiltonians. Heide *et al.* [68] derive a non-separable shadow Hamiltonian for the generalised leapfrog integrator in RMHMC, which results in improved performance relative to sampling from the true Hamiltonian. These methods still suffer from the practical impediment of setting the integration step size and trajectory length.

Hoffman and Gelman [73] present the NUTS methodolog which automatically sets the trajectory length of HMC by setting a termination criterion that avoids retracing of

steps. Hoffman and Gelman [73] also address step size adaptation through primal-dual averaging in the burn-in phase. These two approaches for setting these parameters have been widely adopted in the literature [167, 16, 30, 110].

There has been no attempt to adaptively set the step size and trajectory length parameters for shadow HMC methods. A particular constraint in trajectory length tuning is that most shadow Hamiltonian methods proposed in the literature rely on non-separable Hamiltonians. These non-separable Hamiltonian require special treatment of the auxiliary momentum variable as the momentum is no longer Gaussian. This precludes these methods from being directly used within the NUTS methodology [4, 140, 77, 68]. We overcome this constraint by relying on a separable shadow Hamiltonian based sampler, which is the S2HMC algorithm. This allows for both tuning of step sizes through primal-dual averaging and the trajectory length via a binary tree recursion as in Hoffman and Gelman [73]. We refer to this new algorithm as adaptive S2HMC.

The sampling performance of the proposed adaptive S2HMC method is compared against NUTS, S2HMC with a fixed trajectory length and S2HMC with a uniform random trajectory length. We refer to the latter approach that uses a random trajectory length as Jittered S2HMC (JS2HMC). Note that the step size is tuned via primal-dual averaging during the burn-in period in all the methods. We show that our adaptive S2HMC method achieves better exploration of the posterior and higher effective sample sizes than S2HMC, JS2HMC and NUTS across various benchmarks while leaving the target density invariant. The analysis is performed on the Banana shaped distribution, a multivariate Gaussian distribution with $D = 10$, Neal's [123] funnel with $D = 25$ and BLR posterior targets using the Pima and Australian credit datasets as set out in Table 3.2.

The adaptive S2HMC method differs from NUTS in that a different integrator is used. Instead of using the leapfrog integrator associated with HMC, the processed leapfrog integrator, which is explicit and symplectic, corresponding to S2HMC is used. The processed leapfrog integrator used in the adaptive S2HMC sampler proceeds by first performing the pre-processing step and then passing the momenta, position, and the shadow density to the leapfrog integrator, after which a post-processing step is employed to recover the momenta and position. The post-processed position and momenta are then

Algorithm 13 Adaptive Separable Shadow Hybrid Hamiltonian Monte Carlo Algorithm

Main loop same as NUTS in Algorithm 6. We are extending the BuildTree method in Algorithm 7 to support the processed leapfrog integration scheme.

function BuildTree$(w, p, u, v, j, \epsilon, w^0, p^0)$

Output: $w^-, p^-, w^+, p^+, w', p', n', s', \alpha', n'_\alpha$

if $j = 0$ **then**
  Apply the pre-processing mapping $(\hat{w}, \hat{p}) = \mathcal{X}(w, p)$ in equation (2.33)
  $w^*, p^* =$ **Leapfrog**$(\hat{p},\ \hat{w},\ \epsilon v,\ \tilde{H})$
  Apply the post-processing mapping $(w', p') = \mathcal{X}^{-1}(w^*, p^*)$ in equation (2.34)
  $n' = \mathbf{I}[u < \exp(-\tilde{H}(w', p')]$, $s' = \mathbf{I}[u < \exp(\Delta_{max} - \tilde{H}(w', p')]$
  $n_\alpha = 1$, $\alpha = \min\left(1, \exp(\tilde{H}(w^0, p^0) - \tilde{H}(w', p'))\right)$
  return $w', p', w', p', w', p', n', s', \alpha, n_\alpha$
else
  $w^-, p^-, w^+, p^+, w', p', n', s', \alpha', n'_\alpha =$ BuildTree$(w, p, u, v, j-1, \epsilon, w^0, p^0)$
  if $s' = 1$ **then**
    if $v = -1$ **then**
      $w^-, p^-, -, -, w'', p'', n'', s'', \alpha'', n''_\alpha =$ BuildTree$(w^-, p^-, u, v, j-1, \epsilon, w^0, p^0)$
    else
      $-, -, w^+, p^+, w'', p'', n'', s'', \alpha'', n''_\alpha =$ BuildTree$(w^+, p^+, u, v, j-1, \epsilon, w^0, p^0)$
    end if
    With prob. $\frac{n'}{n'+n''}$, set $w' = w''$, $p' = p''$
    $\alpha' = \alpha' + \alpha''$, $n' = n' + n''$, $n'_\alpha = n'_\alpha + n''_\alpha$
    $s' = s''\mathbf{I}[(w^+ - w^-)p^- \geq 0]\mathbf{I}[(w^+ - w^-)p^+ \geq 0]$
  end if
  return $w^-, p^-, w^+, p^+, w', p', n', s', \alpha', n'_\alpha$
end if

used in the stopping criterion to prevent a U-turn. Note that the stopping criteria in

adaptive S2HMC are the same as in the original NUTS algorithm, except that now the target distribution is the shadow density, and the position and momentum are the post-processed position and momentum. That is, the criterion in equation (2.39) changes from using $\pi_{\mathbf{Z}}(\mathbf{z})$ to using the shadow equivalent $\hat{\pi}_{\mathbf{Z}^*}(\mathbf{z}^*)$ where $\mathbf{z}^*$ is the state representing the post-processed position and momenta. We then calculate the weights of the adaptive S2HMC method, as it is an importance sampler, by comparing the modified and true Hamiltonian at each generated state. Theorem 7.2.1 guarantees that the new method leaves the target distribution, which is the modified density, invariant.

Theorem 7.2.1. *Adaptive S2HMC satisfies detailed balance and thus leaves the target distribution invariant.*

Proof. To show that adaptive S2HMC satisfies detailed balance, we are required to show that the processed leapfrog algorithm is both symplectic and reversible as necessitated by the NUTS methodology. This is guaranteed by Theorem 2.7.2, with the proof presented in Section 2.7. □

We present the pseudo-code for the adaptive S2HMC sampler in Algorithm 13, with parts in blue indicating the additions to NUTS with primal-dual averaging methodology of Hoffman and Gelman [73]. Suppose the blue sections are omitted, and we now sample from the true Hamiltonian. In that case, the processed leapfrog reduces to the original leapfrog integration scheme, which then makes adaptive S2HMC to become NUTS. Note that when the dash "$-$" character is used as an output, it means that the corresponding argument is left unaltered.

For JS2HMC, we generate each sample using a random trajectory length drawn from a discrete uniform distribution. That is, $L \sim U(1, L_{max})$ with L_{max} specified by the user. The use of random trajectory lengths has been explored before in the context of HMC [71, 164], but is yet to be considered for S2HMC. To show that JS2HMC preserves detailed balance, one would need to note that JS2HMC amounts to using a mixture of different S2HMC transition kernels which each preserve detailed balance, and hence JS2HMC preserves detailed balance. JS2HMC serves as a naive adaptive scheme for S2HMC, which should highlight if the more advanced adaptive schemes are adding any value.

7.3 Experiment Description

We consider five test problems to demonstrate the performance of adaptive S2HMC over S2HMC, JS2HMC and NUTS. The targets are the Banana shaped distribution, a multivariate Gaussian distribution with $D = 10$, Neal's [123] funnel with $D = 25$ and BLR posterior targets using the Pima and Australian credit datasets as set out in Table 3.2.

Performance is measured via ESS and ESS per second, and we assess the convergence behaviour of the chains using the $\hat{R}$ metric. For all the algorithms, the step size was set by targeting an acceptance rate of 80% during the burn-in period. The trajectory length for the targets considered in this chapter was set to the values outlined in Table 3.8 for HMC. Ten independent chains were run for each method on each target distribution. Ten thousand samples were generated for each target, with the first 5 000 samples discarded as burn-in. These settings were sufficient for all the algorithms to converge on all the targets.

7.4 Results and Discussion

Figure 7.1 shows the diagnostic trace-plots of the negative log-likelihood across various target posteriors. The results show that all four methods have converged on the target densities considered.

The performance of the algorithms across different metrics is shown in Figure 7.2 and Tables 7.1 to 7.4. In Figure 7.2, the plots on the first row for each dataset show the effective sample size, and the plots on the second row show the effective sample size normalised by execution time. The results are for the ten runs of each algorithm. The execution time t in Figure 7.2 and Tables 7.1 to 7.4 is in seconds. The results in Tables 7.1 to 7.4 are the mean results over the ten runs for each algorithm. We use the mean values over the ten runs in Tables 7.1 to 7.4 to form our conclusions about the performance of the algorithms.

The results show that NUTS and adaptive S2HMC produce similar average trajectory lengths L across the targets, but with adaptive S2HMC producing the larger step size on the majority of the targets. Note that the average trajectory lengths L are calculated

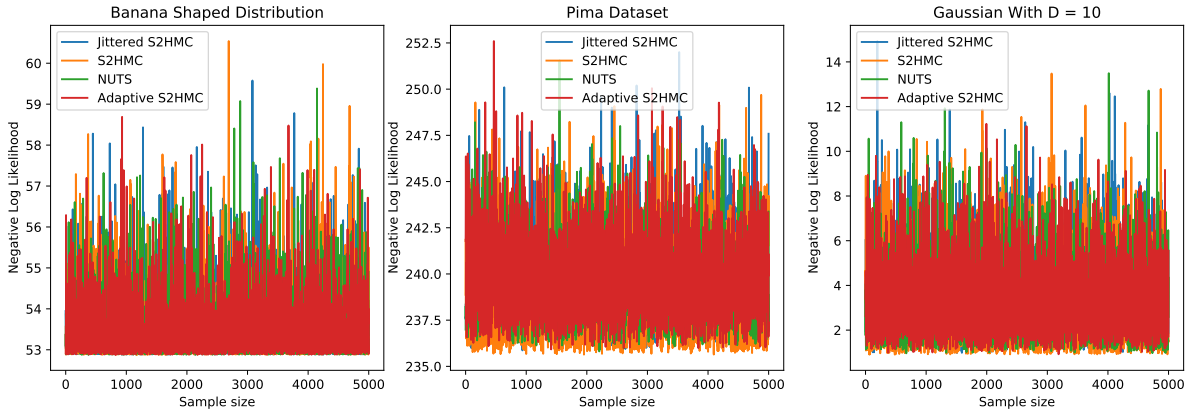


Figure 7.1: Diagnostic trace-plots of the negative log-likelihood across various targets averaged over ten runs of each method. These results show that all the MCMC methods have converged on all the targets.

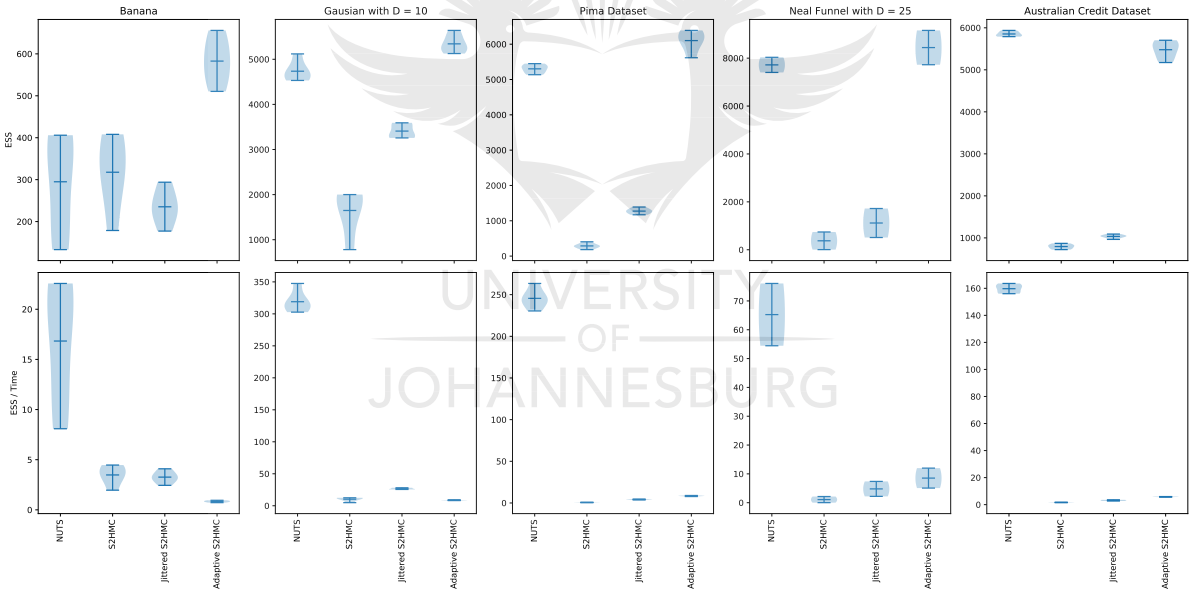


Figure 7.2: Results for the datasets over ten runs of each method. For each dataset, the plots on the first row show the multivariate effective sample size and the plots on the second row show the multivariate effective sample size normalised by execution time (in seconds). For all the plots, the larger the value, the better the method. The dark horizontal line in each violin plot represents the mean value over ten runs of each algorithm.

Table 7.1: Banana shaped distribution results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. Average L represents the average trajectory length used to generate the post-burn-in samples.

Metric	NUTS	S2HMC	Jittered S2HMC	Adaptive S2HMC
Average L	7	15	7	12
ϵ	0.14	0.12	0.13	0.10
ESS	294	317	235	**582**
t	**17.26**	91.26	72.12	693.37
ESS/t	**16.83**	3.48	3.26	0.84
$\hat{R}$ max	**1.00**	**1.00**	**1.00**	**1.00**

post the burn-in period. This combination of step size and trajectory length result in substantial execution times for adaptive S2HMC, which can be attributed to the multiple times the shadow density is evaluated as the adaptive algorithm moves forwards and backwards in time to ensure detailed balance. NUTS outperforms all the methods on an execution time basis, followed by JS2HMC. Note that JS2HMC uses, on average, half the trajectory length of S2HMC due to drawing the trajectory length from a uniform distribution, where the mean would be the upper bound divided by two. In our case, we set the upper bound to be equal to the trajectory length used in S2HMC. We also find that S2HMC and JS2HMC produce similar step sizes, with JS2HMC producing the largest step size of all the methods across all the target posteriors.

We find that adaptive S2HMC outperforms all the methods on an ESS basis across all the targets except for the Australian credit dataset, where it is outperformed by NUTS. More crucially, the adaptive S2HMC produces significantly higher ESSs when compared to S2HMC. What is interesting to note about the ESS performance of S2HMC is that when we target an acceptance rate of 80% as done in this chapter, it produces lower ESSs than when using the step size of HMC in Table 3.8 (as we saw in Chapter 5), which was a smaller step size. The drop in ESS is attributed to non-uniform weights that tend to be produced when we target lower acceptance rates for the generated samples. This suggests that S2HMC prefers targeting higher acceptance rates; this is to ensure that the resultant

Table 7.2: Multivariate Gaussian distribution with $D = 10$ results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. Average L represents the average trajectory length used to generate the post-burn-in samples.

Metric	NUTS	S2HMC	Jittered S2HMC	Adaptive S2HMC
Average L	6	15	7	7
ϵ	0.38	0.48	0.48	0.40
ESS	4 735	1 648	3 408	**5 342**
t	**14.84**	160.63	127.64	628.53
ESS/t	**318.99**	10.26	26.70	8.50
$\hat{R}$ max	**1.00**	**1.00**	**1.00**	**1.00**

weights of the importance sampler are uniform. JS2HMC outperforms S2HMC on an ESS basis across all the majority of the targets, suggesting that one can achieve more uniform weights in S2HMC by simply making the trajectory length random. This naive approach requires minimal modifications to already existing S2HMC implementations.

The adaptive S2HMC method outperforms S2HMC and JS2HMC on a normalised ESS basis on all the targets except on the Banana shaped distribution and multivariate Gaussian distribution. JS2HMC outperforms S2HMC across all the targets except on the Banana shaped distribution. Given the very insignificant change that is required to create JS2HMC from S2HMC, there seems to be little reason not to use it instead of S2HMC in practice. Overall, NUTS outperforms all the methods across all the targets on a time-normalised ESS basis, and all the methods produced good convergence on all the targets as measured by the $\hat{R}$ metric.

Table 7.3: Neal's funnel density with $D = 25$ results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. Average L represents the average trajectory length used to generate the post-burn-in samples

Metric	NUTS	S2HMC	Jittered S2HMC	Adaptive S2HMC
Average L	20	50	25	17
ϵ	0.30	0.30	0.33	0.017
ESS	7 719	374	1 116	**8 441**
t	**120.80**	347.78	233.14	1 143.33
ESS/t	**65.25**	1.07	4.78	8.53
$\hat{R}$ max	**1.00**	**1.00**	**1.00**	**1.00**

Table 7.4: BLR dataset results results averaged over ten runs. The time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric. Average L represents the average trajectory length used to generate the post-burn-in samples

Pima Dataset				
Metric	NUTS	S2HMC	Jittered S2HMC	Adaptive S2HMC
Average L	7	50	25	7
ϵ	0.10	0.13	0.14	0.11
ESS	5 302	288	1 273	**6 102**
t	**21.62**	485.28	306.41	724.87
ESS/t	**245.53**	0.59	4.16	8.42
$\hat{R}$ max	**1.00**	**1.00**	**1.00**	**1.00**
Australian Credit Dataset				
Average L	10	50	25	8
ϵ	0.10	0.15	0.16	0.14
ESS	5 853	794	1 034	**5 478**
t	**36.65**	484.87	335.68	943.27
ESS/t	**159.73**	1.63	3.11	5.80
$\hat{R}$ max	**1.00**	**1.00**	**1.00**	**1.00**

7.5 Conclusion

In this chapter, we introduced the novel adaptive S2HMC algorithm, which combines the benefits of sampling from a shadow Hamiltonian and adaptively sets the trajectory length and step size parameters for S2HMC, while leaving the target invariant. The new method was compared against S2HMC, JS2HMC and NUTS. The JS2HMC method utilises a random trajectory length, and we are the first to introduce such a concept for shadow Hamiltonians into the literature. The empirical results show that the new algorithms provide significant improvement on the S2HMC algorithm.

An important limitation of the adaptive S2HMC is the computational time associated with the method. This leads to poor performance on a normalised ESS basis. We aim to address this issue in the future by using a surrogate model as we intend to do for SMHMC outlined in Chapter 6. The naive JS2HMC method offers improvements to S2HMC without requiring much extra effort, and at half the execution time of S2HMC. It thus provides a practical adaptive scheme for non-expert users.

In the following chapter, we rely on the coupling theory of Hamiltonian samplers and incorporate antithetic sampling into the MHMC and S2HMC samplers. This approach leads to a reduction of the variance of their estimators or, equivalently, serves to increase their effective sample sizes.

Chapter 8

Antithetic Hamiltonian Monte Carlo Techniques

8.1 Introduction

In the preceding chapters, we proposed novel extensions to the MHMC and S2HMC algorithms that enhance their performance through the utilisation of random mass matrices, partial momentum retention, automatically tuning their parameters as well as the use of integrator-dependent shadow Hamiltonians. In this chapter, we use antithetic sampling to reduce the variance of Hamiltonian dynamics-based estimators and consequently increase the ESSs of their samplers. A-HMC has been recently introduced into literature by Piponi *et al.* [136] and has been shown to outperform antithetic versions of random walk MH and MALA. We now introduce novel antithetic samplers based on the MHMC and S2HMC methods to create samplers that outperform A-HMC. The results show that these two new methods outperform their non-antithetic counterparts as well as A-HMC on a time-normalised ESS basis. The results also indicate the overall utility that can be derived from incorporating antithetic sampling in Hamiltonian dynamics-based MCMC methods. The material in this chapter was published in the following journal article:

- **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Antithetic Magnetic and Shadow Hamiltonian Monte Carlo.* IEEE Access, vol. 9, pp. 49857-49867.

8.2 Proposed Antithetic Samplers

In their inspired work, Piponi *et al.* [136] combined antithetic sampling and control variate variance reduction techniques with Hamiltonian dynamics to create the A-HMC and control variate HMC methods respectively. The HMC based antithetic samplers

Algorithm 14 Antithetic Magnetic Hamiltonian Monte Carlo Algorithm

Input: N, G, ϵ, L, w^x_{init}, w^y_{init}, $H(w,p)$

Output: $(w^x)_{m=0}^N$, $(w^y)_{m=0}^N$

$w^x_0 \leftarrow w^x_{\text{init}}$

$w^y_0 \leftarrow w^y_{\text{init}}$

for $m \rightarrow 1$ **to** N **do**

  $p^x_{m-1} \sim \mathcal{N}(0, \mathbf{M})$

  $p^y_{m-1} = -p^x_{m-1} \leftarrow$ **Momentum shared between the two chains**

  $p^x_m,\, w^x_m =$ **Integrator**$(p^x_{m-1},\, w^x_{m-1},\, \epsilon,\, L,\, \text{G},\, H)$

  $p^y_m,\, w^y_m =$ **Integrator**$(p^y_{m-1},\, w^y_{m-1},\, \epsilon,\, L,\, \text{G},\, H)$

  $\delta H^x = H(w^x_{m-1}, p^x_{m-1}) - H(w^x_m, p^x_m)$

  $\delta H^y = H(w^y_{m-1}, p^y_{m-1}) - H(w^y_m, p^y_m)$

  $\alpha^x_m = \min\left(1, \exp\left(\delta H^x\right)\right)$

  $\alpha^y_m = \min\left(1, \exp\left(\delta H^y\right)\right)$

  $u_m \sim \text{Unif}(0,1) \leftarrow$ **Uniform random number shared between the two chains**

  The Metropolis-Hastings step is the same as in Algorithm 3 for both chains.

end for

have an advantage over antithetic Gibbs samplers presented in Frigessi *et al.* [48] in that they are applicable to problems where conditional distributions are intractable, and where Gibbs sampling may mix slowly [136]. The control variate HMC variance

Algorithm 15 Antithetic Separable Shadow Hamiltonian Hybrid Monte Carlo Algorithm

Input: N, ϵ, L, w^x_{init}, w^y_{init}, $H(w,p)$, $\tilde{H}(w,p)$

Output: $(w^x)_{m=0}^N$, $(w^y)_{m=0}^N$, $(b^x)_{m=0}^N$, $(b^y)_{m=0}^N$

$w_0^x \leftarrow w^x_{\text{init}}$

$w_0^y \leftarrow w^y_{\text{init}}$

for $m \to 1$ **to** N **do**

  $p^x_{m-1} \sim \mathcal{N}(0, \mathbf{M})$

  $p^y_{m-1} = -p^x_{m-1}$ **Momentum shared between the two chains**

  Apply the pre-processing mapping to both chains

  $p^x_m, w^x_m =$ **Leapfrog**$(p^x_{m-1}, w^x_{m-1}, \epsilon, L, \tilde{H})$

  $p^y_m, w^y_m =$ **Leapfrog**$(p^y_{m-1}, w^y_{m-1}, \epsilon, L, \tilde{H})$

  Apply the post-processing mapping to both chains

  $\delta H^x = \tilde{H}(w^x_{m-1}, p^x_{m-1}) - \tilde{H}(w^x_m, p^x_m)$

  $\delta H^y = \tilde{H}(w^y_{m-1}, p^y_{m-1}) - \tilde{H}(w^y_m, p^y_m)$

  $\alpha^x_m = \min\left(1, \exp\left(\delta H^x\right)\right)$

  $\alpha^y_m = \min\left(1, \exp\left(\delta H^y\right)\right)$

  $u_m \sim \text{Unif}(0,1) \leftarrow$ **Uniform random number shared between the two chains**

  The Metropolis-Hastings step and the weight calculation is the same as in Algorithm 5 for both chains.

end for

reduction technique is more generally applicable than the antithetic approach, which requires the target distribution to be symmetric about some vector. Control variate techniques rely on the practitioner being able to construct efficient and accurate approximate distributions, which is difficult for neural networks and is an impediment to

general use [136]. Thus, this thesis focuses on the antithetic sampling approach, which is straightforward to implement for BNNs as well as for BLR. In particular, we expand on the work of Piponi *et al.* [136] by presenting two new antithetic sampling MCMC methods being the Antithetic Separable Shadow Hamiltonian Hybrid Monte Carlo (A-S2HMC) and Antithetic Magnetic Hamiltonian Monte Carlo (A-MHMC) algorithms.

The A-S2HMC algorithm is based on sampling from the shadow Hamiltonian using S2HMC [77, 155]. Sampling using the S2HMC algorithm provides benefits over HMC - and the results in the preceding chapters of this thesis attest to this. One benefit is that the shadow Hamiltonian is better conserved by the numerical integrator, which allows one to use larger step sizes, and thus reducing auto-correlations, than in HMC without a significant drop in the acceptance rates [77, 155, 140, 4]. In addition, S2HMC is an importance sampler and already offers variance reduction over HMC. The A-S2HMC algorithm thus combines the benefits of antithetic sampling with importance sampling, which should provide even more variance reduction than is provided by S2HMC over HMC. The disadvantage of S2HMC over HMC is that it consumes more computational resources than HMC, which reduces the outperformance on a time-normalised ESS basis.

The A-MHMC algorithm is based on adding antithetic sampling to the MHMC algorithm. The MHMC algorithms provide an improvement on HMC by adjusting the range of exploration via a magnetic field. This has the effect of enhancing the convergence speed and reducing the auto-correlations of the samples [156, 59]. In the A-MHMC algorithm, we combine antithetic sampling with the benefits that MHMC already has over HMC intending to create a sampler that outperforms A-HMC. Unlike S2HMC, MHMC has comparable execution time with HMC, which means that the computational burden should not outweigh the benefits of higher ESSs produced by MHMC and A-MHMC, respectively.

The pseudo-code for the new antithetic algorithms that we are proposing is presented in Algorithms 14 and 15. The difference between the antithetic algorithms and the original algorithms is that the momentum variable and the uniform random variable in the MH acceptance step are shared between the two chains. These differences are appropriately highlighted in the respective algorithms.

8.3 Experiment Description

The performance of the algorithms is compared on real-world benchmark datasets. The real-world data used are the four classification datasets used in Girolami and Calderhead [53], which we model using BLR. We also apply BNNs to model real world regression benchmark datasets. The datasets, their number of features and their associated models are outlined in Table 3.2. For all the algorithms, we set the mass matrix $\mathbf{M} = \mathbf{I}$, which is the common approach in practice [17, 124].

The performance metrics used in this chapter are the multivariate ESS, the execution time and the ESS normalised by the execution time. Note that the ESS for the variance reduced chain is related to the ESS of the original chain as [136]:

$$ESS_{antithetic} = \frac{2 \times ESS_{original}}{1 + \eta} \tag{8.1}$$

where η is the correlation coefficient between the corresponding pairs of chains. In this thesis, the correlation is taken as the maximum correlation across all the parameter dimensions, which creates a lower bound for $ESS_{antithetic}$. Note that we estimate η using the Spearman rank correlation coefficient [170]. A careful analysis of equation (8.1) reveals that we can increase the ESS by ensuring that the correlation is as close as possible to -1, which is what the anti-coupling methodology aims to do. This also means that the $ESS_{antithetic}$ can be orders of magnitude greater than the total number of generated samples N.

For all the datasets, we generated three thousand samples with a burn-in period of one thousand samples. This was sufficient for all the algorithms to converge on all the datasets. The step sizes were chosen by targeting an acceptance rate of 95%, as suggested by Piponi *et al.* [136], through the primal-dual averaging methodology. Note that we only tune the step size of the primary chain X, and use this step size in the secondary chain Y. The trajectory length parameters for each target are the same as the ones in Table 3.8, with S2HMC using the same trajectory length as HMC, and the antithetic versions using the same trajectory length as the non-antithetic counterparts. In evaluating the S2HMC algorithm and its antithetic variant, we set a convergence tolerance of 10^{-6} or the completion of one-hundred fixed point iterations.

8.4 Results and Discussion

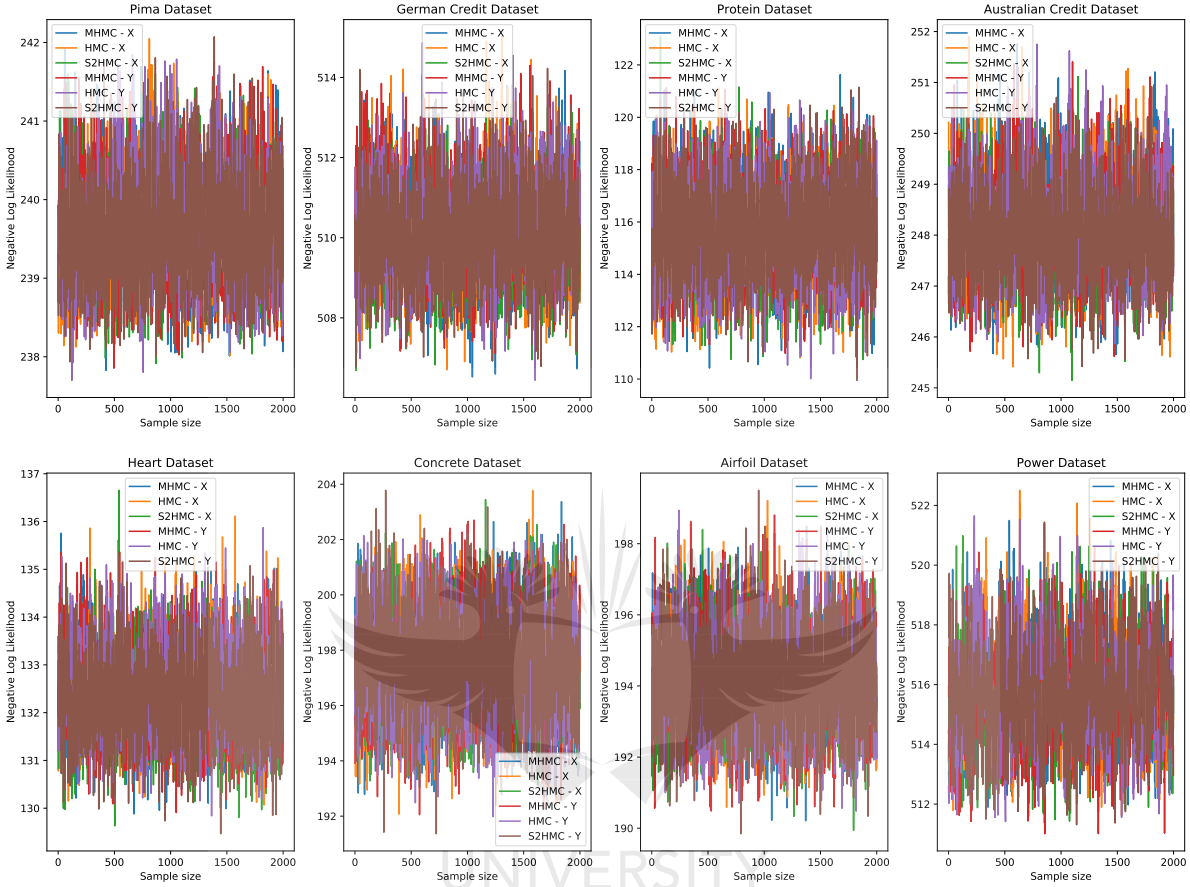


Figure 8.1: Diagnostic trace-plots of the negative log-likelihood across various targets averaged over ten runs of each method. These results show that all the MCMC methods have converged on all the targets. X is the primary chain and Y is the secondary chain for each method.

Figure 8.1 shows the diagnostic trace-plots of the negative log-likelihood averaged over ten runs for both the primary (X) and secondary/antithetic (Y) chains. The results show that all the MCMC methods have converged on all the targets, and on both the X and Y chains.

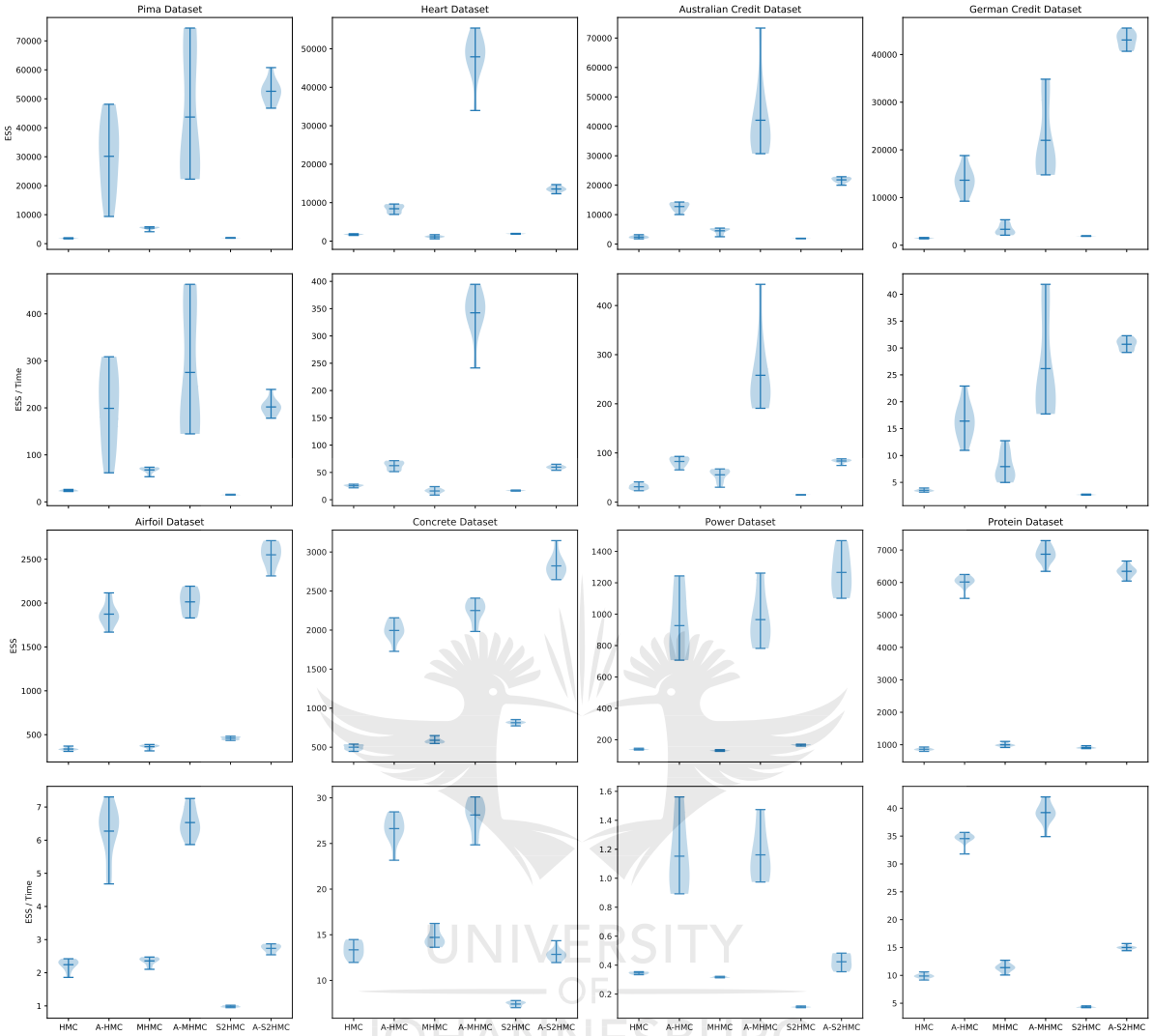


Figure 8.2: Results for the datasets over ten runs of each method across the BLR and BNN datasets. For each dataset, the plots on the first row show the multivariate effective sample size and the plots on the second row show the multivariate effective sample size normalised by execution time (in seconds). For all the plots, the larger the value, the better the method. The dark horizontal line in each violin plot represents the mean value over ten runs of each algorithm.

The performance of the algorithms across different metrics is shown in Figure 8.2 and Tables 8.1 and 8.2. In Figure 8.2, the plots on the first row of each dataset show the ESS, and the plots on the second row show the ESS normalised by execution time

Table 8.1: Mean results over ten runs of each algorithm for the BLR datasets. Each column represents the mean value for the specific method. The execution time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric.

Pima Indian dataset						
	HMC	A-HMC	MHMC	A-MHMC	S2HMC	A-S2HMC
ϵ	0.0577		0.0076		**0.1152**	
ESS	1 831	30 152	5 367	43 703	1 970	**52 595**
t (in secs)	**76**	151	79	157	130	260
ESS/t	24.13	198.96	67.99	**275.38**	15.12	201.92
η	-0.8462		-0.6967		**-0.9247**	
Heart dataset						
ϵ	0.08307		0.0058		**0.17799**	
ESS	1 721	8 418	1 119	**47 917**	1 912	13 545
t (in secs)	**67**	134	70	139	114	227
ESS/t	25.53	62.45	16.01	**342.41**	16.84	59.63
η	-0.5884		**-0.9536**		-0.7172	
Australia credit dataset						
ϵ	0.0546		0.0068		**0.1213**	
ESS	2 405	12 751	4 515	**42 075**	1 925	21 820
t (in secs)	**77**	154	81	162	130	260
ESS/t	31.178	82.64	55.45	**257.97**	14.76	83.63
η	-0.6209		-0.7624		**-0.8232**	
German credit dataset						
ϵ	0.0295		0.0074		**0.0772**	
ESS	1 465	13 630	3 335	22 000	1 899	**43 045**
t (in secs)	**416**	832	420	840	701	1 402
ESS/t	3.52	16.40	7.93	26.16	2.70	**30.69**
η	-0.7779		-0.6898		**-0.9116**	

Table 8.2: Mean results over ten runs of each algorithm for the BNN datasets. Each column represents the mean value for the specific method. The execution time t is in seconds. The values in **bold** indicate that the particular method outperforms the other methods on that specific metric.

Power dataset						
	HMC	A-HMC	MHMC	A-MHMC	S2HMC	A-S2HMC
ϵ	0.0151		0.0115		**0.0307**	
ESS	138	927	131	965	166	**1 266**
t (in secs)	**402**	804	414	831	1 499	2 991
ESS/t	0.35	1.15	0.31	**1.16**	0.11	0.42
η	-0.6873		-0.7218		**-0.7332**	
Airfoil dataset						
ϵ	0.0290		0.0278		0.0627	
ESS	336	1 874	363	2 013	459	**2 550**
t (in secs)	**149**	301	154	308	466	932
ESS/t	2.24	6.27	2.36	**6.53**	0.98	2.73
η	-0.6397		-0.6384		**-0.6392**	
Concrete dataset						
ϵ	0.0366		0.0367		**0.0935**	
ESS	500	1 995	588	2 250	813	**2 822**
t (in secs)	**37**	74	40	80	110	219
ESS/t	13.34	26.63	14.70	**28.12**	7.41	12.85
η	**-0.4984**		-0.4766		-0.4226	
Protein dataset						
ϵ	0.0437		0.0438		**0.0790**	
ESS	860	6 013	997	**6 870**	914	6 343
t (in secs)	**87**	174	88	175	211	422
ESS/t	9.88	34.54	11.39	**39.22**	4.32	14.99
η	**-0.7138**		-0.7096		-0.711	

(which is in seconds). The results are for the ten runs of each algorithm. As with Figure 8.2, the execution time t in Tables 8.1 and 8.2 is in seconds. The results in Tables 8.1 and 8.2 are the mean results over the ten runs for each algorithm. Note that we use the mean values over the ten runs in Tables 8.1 and 8.2 to form our conclusions about the performance of the algorithms.

We find that the shadow Hamiltonian methods produce the largest step size for the targeted 95% acceptance rate. This can be attributed to the leapfrog integration scheme better conserving the modified Hamiltonian compared to the true Hamiltonian. This has been a recurring observation in this thesis. Furthermore, we find that the MHMC methods produce the smallest step size. This is due to the presence of the magnetic field, which has the effect of reducing the step size required to reach the same level of sample acceptance rates when compared to HMC. This phenomenon is also clearly highlighted in Table 3.8 where MHMC is shown to produce lower step sizes consistently across all the target posterior distributions.

On the majority of the datasets, we find that MHMC and S2HMC have higher ESSs than HMC, with the outperformance being more pronounced on the BLR datasets. The A-MHMC and A-S2HMC methods produce superior ESSs on all 8 benchmark datasets when compared to A-HMC. A-MHMC produces higher ESSs than A-HMC and A-S2HMC on 7 of the 8 benchmark datasets, with A-MHMC underperforming A-S2HMC on the German credit dataset. These results show that the proposed methods produce significant outperformance on the BLR datasets, with the outperformance on the BNN datasets being less than on the BLR datasets. An explanation of this behaviour can be traced to two aspects: 1) the average correlations η produced by the methods - the proposed method produce very high correlations on the BLR datasets, with A-HMC producing comparable or better average correlations to MHMC and S2HMC on the BNN datasets, and 2) the magnetic field used for the MHMC methods may not be optimal for the BNN datasets, and this can be seen by the marginal outperformance of MHMC over HMC on an ESS basis on the BNN datasets.

As expected, HMC has the lowest execution time t on all of the datasets, with MHMC being a close second. The execution time for MHMC is similar to that of HMC on the BLR datasets and becomes slightly worse on the BNN datasets. This can be attributed

to the larger dimensionality of the BNN models when compared to BLR models, which pronounces the computational costs of the extra matrix operations required for MHMC. The large compute time of the shadow Hamiltonian algorithms can be attributed to the fixed-point iterations in equations (2.33) and (2.34). Note that the execution time for the antithetic variants is twice the execution time for the non-antithetic algorithms. This is because two chains are run for the antithetic versions. The execution time of all the algorithms increases with the dimensionality of the problem, which is expected. On a time-normalised ESS basis, A-MHMC outperforms A-HMC on all the datasets.

The results in this chapter show that although the proposed methods have, in general, a higher computational cost than HMC, they still provide improved normalised ESS rates. It is also worth noting that all the algorithms have similar predictive performance on all the benchmark datasets, with the predictive performance being similar between the antithetic and non-antithetic counterparts.

8.5 Conclusion

In this chapter, we introduced the antithetic S2HMC and antithetic MHMC variance reduction schemes based on approximate Markov chain coupling. We compare these two new algorithms to the A-HMC algorithm on classification tasks using BLR, and on regression tasks using BNN models. We find that the antithetic versions of all the algorithms have higher ESSs than their non-antithetic variants, which shows the usefulness of antithetic MCMC methods.

The work in this chapter can be improved by automatically tuning the magnetic term in MHMC and A-MHMC, which should improve the results on the BNN datasets. We plan to consider larger datasets and deeper neural networks in future work, albeit the computation cost will be higher. A comparison of the performance of the proposed algorithms to their control variate counterparts, and the exploration of the possibility of using Riemannian manifold-based Monte Carlo algorithms is also of interest.

In the following chapter, we present the first application of MCMC methods to the modelling of financial statement audit outcomes, with a focus on South African local government entities.

Chapter 9

Bayesian Inference of Local Government Audit Outcomes

9.1 Introduction

This chapter presents the first-in-literature application of Bayesian inference to predict financial statement audit outcomes of South African local government entities. The audit outcomes are modelled using BLR with financial ratios as input features. The inference is performed using the MHMC, S2HMC, random walk MH, MALA and HMC methods. We employ BLR with ARD to identify the features that are most important when modeling audit outcomes. Most of the considered algorithms agree on which financial ratios are the most relevant for modeling audit opinions. Our analysis shows that the *repairs and maintenance as a percentage of total assets ratio, current ratio, debt to total operating revenue, net operating surplus margin* and *capital cost to total operating expenditure ratio* are the five most important features when predicting local government audit outcomes. These results could be of use for various stakeholders, including the AG-SA, as focusing on these ratios can speed up the detection of fraudulent behaviour in municipal entities and improve the speed and overall quality of the audit. The material in this chapter has been published in the following international journal article:

- **Mongwe, W.T.**, Mbuvha, R. and Marwala, T., 2021. *Bayesian Inference of Local Government Audit Outcomes*. Plos One, doi: 10.1371/journal.pone.0261245

9.2 Background

The AG-SA revealed that South African local government entities lost over $2 billion in irregular expenditure in the 2018-2019 financial year [11, 109]. This irregular expenditure has consequently harmed service delivery and returns on the rapidly increasing government debt [11]. The manipulation of financial statements is not only limited to the public sector, with the Steinhoff collapse being a prime example of management fraud in the private sector [38, 37]. Steinhoff is a South African retailer that lost over R200 billion in market capitalisation on the Johannesburg Stock Exchange (JSE) over a short space of time after allegations of accounting fraud [38, 109].

The recent scandal of Wirecard, and previously Enron, also indicate that financial statement fraud is not only a problem for South Africa, but the world at large [129, 67]. Wirecard is a German payment processing company, which filed for insolvency in 2020, that manipulated its financial statements by misstating its profit [129]. Enron was an American natural gas company that lost over $60 billion in market capitalisation in the early 2000s after the allegations of fraud emerged [67].

The use of automated techniques for the analysis and detection of financial statement fraud has been on the increase in the past decade, as highlighted in Appendix A [109, 107, 45, 54, 61, 82, 90, 106, 134]. In our literature survey of 52 papers, which we summarise in Appendix A, we outline how artificial intelligence and other automated methods can be used to construct decision support tools for various stakeholders. For example, auditors may use the decision support tool to flag entities who are at risk of having committed financial statement fraud and reduce the turnaround time of the audit, amongst other benefits [109, 107]. A prime example within the South African context is the AG-SA having to audit all local and provincial government entities at the end of each financial year [11, 107].

As shown in Table A.3, logistic regression has been successfully used in the literature for the detection of financial statement fraud [135, 45, 81, 86, 76, 133, 130, 7, 105]. Moepya *et al.* [105] use logistic regression in the detection of fraud in companies listed on the JSE, while Boumediene *et al.* [21] performed a similar study for entities listed in Tunisia. Logistic regression has advantages over more complicated models such as artificial neural networks in that the results are more easily interpretable by the stakeholders,

which is an important consideration when building a decision support tool [107, 105, 82].

In logistic regression, as with any other machine learning model, one has to decide on the input features to use. Correctly selecting the variables to use as inputs for the models is essential because it can influence the performance of the models [97]. Utilising feature selection techniques can improve the model's predictive performance and reduce the model complexity as fewer features would be required [31, 109]. Examples of feature selection methods used in the financial statement fraud detection literature include correlation, t-test, analysis of variance, decision trees, and principal component analysis [109, 21, 105, 34, 154, 33, 46]. In this thesis, we limit ourselves to only using financial ratios to predict financial statement audit outcomes. Thus feature selection in this context amounts to selecting which financial ratios are the most important or relevant for inferring financial statement audit opinions.

In this chapter, we present the first use of BLR with automatic relevance determination (BLR-ARD) for the inference of audit outcomes. The Bayesian approach allows us to measure the uncertainty in our predictions, which gives a sense of how much confidence we have in a particular prediction. The use of ARD allows us to automatically determine which of the input features are the most relevant, with uncertainty measures around these as well [97, 95]. This formulation of the problem results in the model outcomes being more interpretable, allowing stakeholders to understand the model results better.

This chapter's motivation is to understand the financial performance of South African municipalities in terms of audit outcomes, particularly the features or financial ratios that drive these audit outcomes. We approach this problem from a Bayesian perspective as it provides a probabilistically principled framework for predicting and understanding the audit performance of local government entities. This framework also enables us to provide uncertainty levels in the predictions produced by the models and further allows us to automatically identify and rank the most important financial ratios for audit outcome modelling using prior distributions - which is an essential contribution of this chapter.

We train the BLR-ARD model parameters with MCMC methods. This chapter also presents the first use of the random walk MH, HMC, MALA, MHMC, S2HMC algorithms in the training of BLR-ARD models for inference of financial statement audit opinions.

9.3 Experiment Description

We model the local government audit outcomes using BLR due to the simplicity of this model and the fact that it is one of the commonly used methods in the FSF literature as shown in Table A.4. The negative log-likelihood $l(D_x|\mathbf{w})$ function, where $\mathbf{w}$ are the model parameters, associated with logistic regression is given by:

$$l(D_x|\mathbf{w}) = \sum_{i}^{N_{obs}} y_i \log(\mathbf{w}^T x_i) + (1 - y_i)\log(1 - \mathbf{w}^T x_i) \tag{9.1}$$

where D_x is the data and N_{obs} is the number of observations. Thus, the target unnormalised posterior log distribution is given as:

$$\ln p(\mathbf{w}|D_x) = l(D_x|\mathbf{w}) + \ln p(\mathbf{w}|\alpha) + \ln q(\alpha) \tag{9.2}$$

where $\ln p(\mathbf{w}|\alpha)$ is the log of the prior distribution placed on the parameters given the hyperparameters, and $\ln q(\alpha)$ is the marginal distribution of the hyperparameters. We model the parameters $\mathbf{w}$ as having a Gaussian prior with each parameter having zero mean and its own standard deviation α_i. The $\alpha_i's$ are assumed to follow a log-normal distribution with mean zero and variance 1. The α_i indicates how vital the parameter associated with the input feature is. The larger the value of α_i, the more critical the input feature is in predicting the audit outcomes and election results, respectively.

The aim is to infer the parameters $\mathbf{w}$ and hyperparameters α using MCMC methods. In the literature, this problem is typically formulated as a Gibbs sampling scheme, where the hyperparameters are sampled first and then the parameters and so on [121, 97]. The approach taken in this chapter is to jointly infer the parameters $\mathbf{w}$ and hyperparameters α. This approach has the advantage of resulting in a more stable exploration of the posterior, at least for the current dataset. However, it results in the effective parameter space being doubled - which can significantly reduce the sampling time compared to the Gibbs sampling approach.

For each of the random walk MH, MALA, HMC, MHMC and S2HMC algorithms used in this chapter, we generate ten Markov chains of 10 000 samples. The first 5 000 samples were used as the burn-in period, and any required tuning of algorithm parameters was performed during the burn-in period. For the HMC, MHMC and S2HMC algorithms we

set the pre-conditioning mass matrix $\mathbf{M} = \mathbf{I}$, which is the common approach in practice [121, 17].

We target a 25% acceptance rate for random walk MH [145], 60% for MALA [145] and 80% acceptance rates for the remaining MCMC algorithms. A trajectory length of 50 was used for HMC, MHMC and S2HMC. We then assess the performance of the algorithms by generating the trace-plots of the unnormalised target posterior distribution, the ESSs of the generated samples, the ESSs of the generated samples normalised by execution time, and predictive performance on unseen data. Note that the execution time is the time taken to generate the samples after the burn-in period.

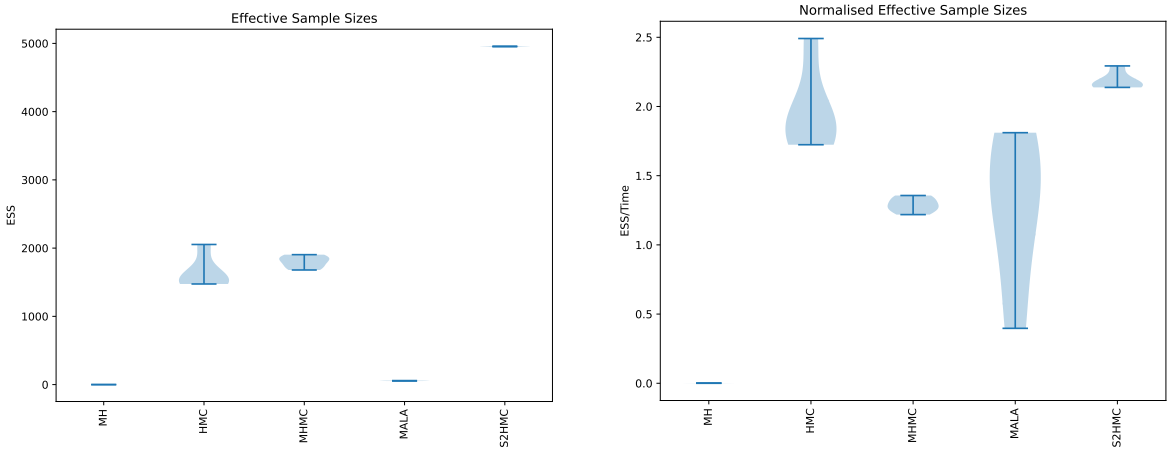
The ESS calculation used is described in Section 3.4.1. The predictive performance on unseen data is performed using the ROC as well as AUC. The ranking of the importance of the financial ratios is performed by calculating the mean or average α, which are the standard deviations in equation (9.2), for each model parameter over the ten chains. The higher the α value, the more important the input financial ratio is for modelling the audit outcomes.

9.4 Results and Discussion

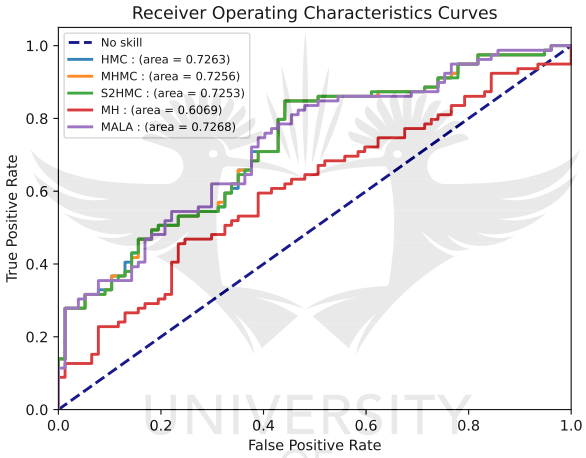
In evaluating the S2HMC algorithm, we set a convergence tolerance of 10^{-6} or the completion of 100 fixed point iterations. Figure 9.1 shows the inference results, while Figures 9.2 and 9.3 shows the input feature importance or relevance for the considered MCMC methods. The results in Figures 9.2 and 9.3 are summarised, as rankings for each financial ratio, in Table 9.1.

Figure 9.1 (a) shows that the S2HMC produces the largest effective sampling sizes, indicating that the algorithm produces less correlated samples when compared to the other methods. HMC and MHMC have the second-highest ESSs, with random walk MH and MALA having very low ESSs, indicating that these two methods produce significantly correlated samples.

Figure 9.1 (b) shows that on a normalised (by execution time) ESS basis, the HMC and S2HMC have similar time normalised ESSs and outperform the other methods. Although MH is also relatively fast, since the ESS it produces is very low, it still under-



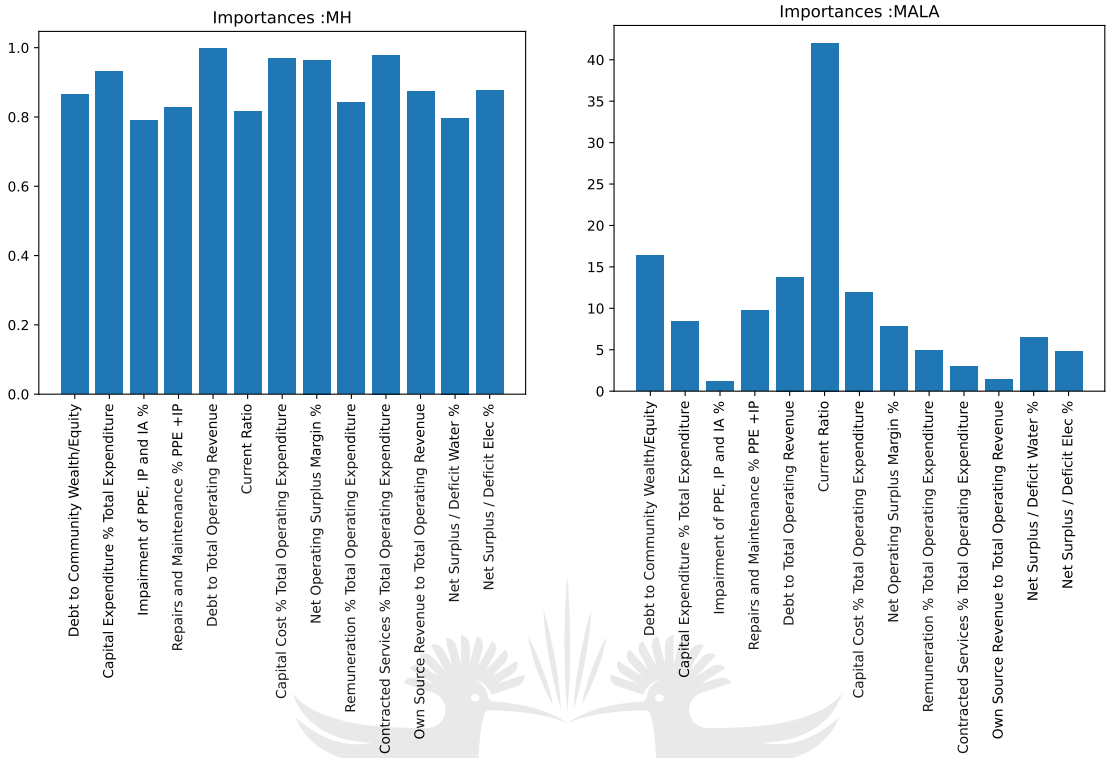
(a) (b)



(c)

Figure 9.1: Inference results for the BLR-ARD model across various sampling methods. a) Effective sample sizes, b) Effective sample sizes normalised by execution time and c) Predictive performance based on the AUC.

performs on a normalised ESS basis. Figure 9.1 (c) shows that the MH algorithm has the lowest predictive performance. S2HMC and MHMC have the joint highest predictive performance, which corresponds with the high ESSs generated by these methods. Note that a convergence analysis of the chains was performed using the $\hat{R}$ metric. We found that all the methods had converged on the target.

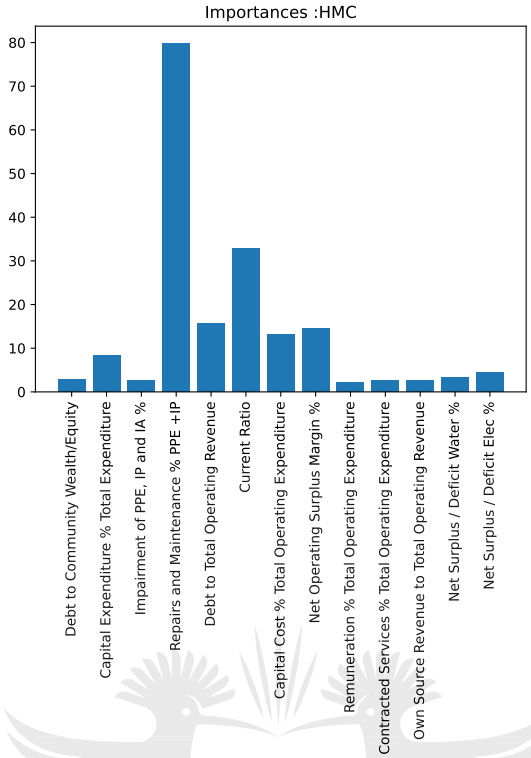


(a) Importance's for MH. (b) Importance's for MALA.

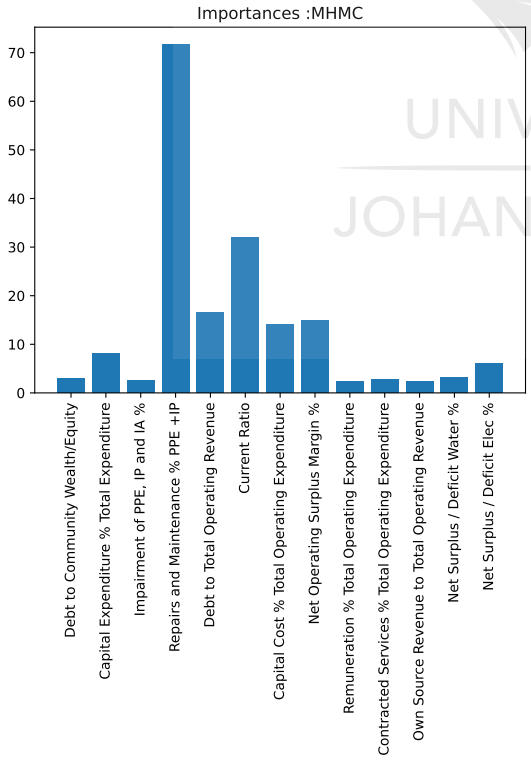
Figure 9.2: Average posterior standard deviations from the random walk MH and MALA algorithms. The higher the value, the more important the financial ratio is to modelling audit opinions.

Figures 9.2 and 9.3 shows the relative importance or relevance of each of the financial ratios produced by each of the MCMC methods. The results show that the MH algorithm struggles to distinguish between important and not-so-important financial ratios. This is because of the poor exploration of the target. On the other hand, the other MCMC methods can extract the importance or most relevant features for the audit opinion modeling task. Table 9.1 shows the ranking of the importance of the financial ratios produced by each of the methods. The most commonly featured financial ratios in the top five rankings are:

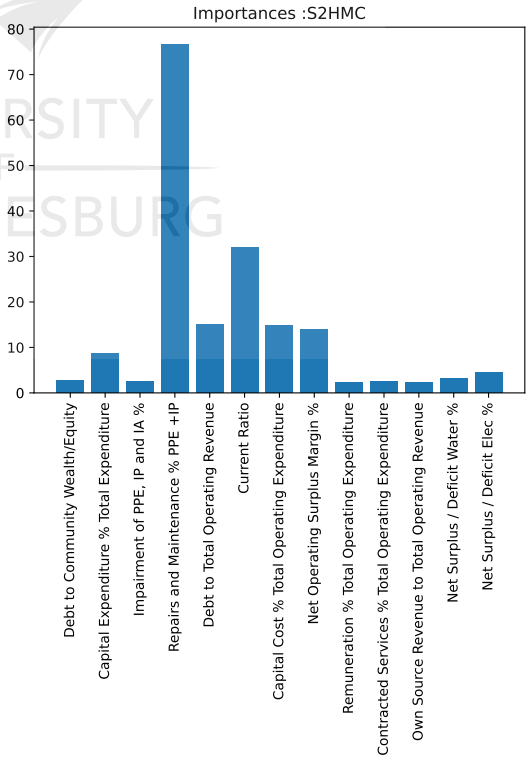
- Ratio 5 - *Debt to Total Operating Revenue*: This ratio is selected by all the five methods



(a) Importance's for HMC.



(b) Importance's for MHMC.



(c) Importance's for S2HMC.

Figure 9.3: Average posterior standard deviations from the HMC, MHMC and S2HMC algorithms. The higher the value, the more important the financial ratio is to modelling audit opinions.

Table 9.1: Ranking of the financial ratios by each method. For example, MHMC ranks ratio four as the most important, while MH ranks ratio seven as the third most important. The financial ratios are described in more detail in Section 3.3.4.

Ranking	MH	MALA	HMC	MHMC	S2HMC
1	5	6	4	4	4
2	10	1	6	6	6
3	7	5	5	5	5
4	8	7	8	8	7
5	2	4	7	7	8
6	13	2	2	2	2
7	11	8	13	13	13
8	1	12	12	12	12
9	9	9	1	1	1
10	4	13	3	10	3
11	6	10	11	3	10
12	12	11	10	11	11
13	3	3	9	9	9

- Ratio 7 - *Capital Cost to Total Operating Expenditure*: This ratio is selected by all the five methods
- Ratio 4 - *Repairs and Maintenance as a percentage of PPE +IP*: This ratio is selected by MALA, HMC, MHMC and S2HMC
- Ratio 6 - *Current Ratio*: This ratio is selected by MALA, HMC, MHMC and S2HMC
- Ratio 8 - *Net Operating Surplus Margin*: This ratio is selected by the MH, MHMC and S2HMC algorithms.

These results are in line with the results we observed in Section 3.3, where features were selected based on the correlation feature selection metric and passed into the SOM. In Section 3.3, we found that the financial ratios that are critical for distinguishing fraudulent instances from non-fraudulent instances are the current ratio, net operating surplus

margin, and the debt to total operating revenue. Our analysis in this chapter shows that the most relevant financial ratios are the repairs and maintenance as a percentage of PPE and IP, followed by the ratios found Section 3.3, with the last relevant ratio being the capital cost to total operating expenditure ratio.

These results make intuitive sense as, for example, a high repairs and maintenance ratio means that the municipality is undertaking more repairs to assets than the assets' total value. This is likely an indication of a lack of adherence to proper corporate governance as repairs to assets should typically be less than the value of those assets - else those assets should be written-off. Furthermore, a high capital cost to total expenditure ratio means that debt repayments are the most significant component of total expenditure, indicating that the entity has a large debt. This might prompt the entity to act in a manner that flouts corporate governance procedures to hide its dire financial situation.

In Section 3.3, we provided an interpretation of the current ratio, net operating surplus margin, and capital cost to total operating expenditure financial ratios in terms of how they relate with audit outcomes. Our findings in this chapter agree with those in Section 3.3 in that we find that a high current ratio, a high net surplus operating margin, and low debt to total operating revenue financial ratios are associated with entities that are less likely to engage in manipulation of their financial statements as they are in good financial standing - with the converse also being true.

These results can prove to be particularly useful for auditors as focusing on these ratios can speed up the detection of inadequate corporate governance behaviour in municipal entities and improve the overall quality of the audits.

9.5 Conclusion

This chapter presented the first-in-literature fully Bayesian approach to inferring financial statement audit opinions. This Bayesian approach is applied to South African local government entity audit outcomes using financial ratios as inputs. The inference is performed using random walk MH, MALA, HMC, MHMC and S2HMC. The sampling was applied to BLR-ARD. Automatic relevance determination allows one to determine which features are the most important in an automated manner and thus implicitly

perform feature selection.

In our approach, the parameters and the hyperparameters, which measure the relevance of the financial ratios, are jointly sampled. The results show that the S2HMC produces the best sampling results, with the largest ESSs. However, the predictive performance of MALA, HMC, MHMC and S2HMC is found to be the same. The random walk MH algorithm produces the worst sampling behaviour due to its random walk nature and has both the lowest effective sample rates and predictive performance.

The results further show that the five most essential features in the modelling of audit outcomes for municipalities are the 1) repairs and maintenance as a percentage of total assets ratio, 2) current ratio, 3) debt to total operating revenue, 4) net operating surplus margin, and 5) capital cost to total operating expenditure ratio. These results could prove helpful for auditors as focusing on these ratios can speed up the detection of possible fraudulent behaviour of municipal entities. The work in this chapter can be improved upon by comparing the performance of the BLR with ARD model with other models such as the BNN with ARD model. Furthermore, we plan to perform this analysis for listed entities and compare the results to the local government entities considered in this chapter. The consideration of a more extensive set of financial ratios could also improve the results.

In the next chapter, we provide a summary of the overall original contributions to knowledge of this thesis as well as ongoing and future work.

Chapter 10

Conclusions

10.1 Summary of Contributions

In this thesis, we explore and propose novel extensions and enhancements to the MHMC and S2HMC algorithms. These methods are descendants of HMC, which is a popular and go-to method for performing Bayesian inference of complex machine learning models. We compare the extensions to MHMC and S2HMC that we introduce to the original algorithms across numerous performance metrics and various benchmark problems. We also present a first-in-literature application of MCMC methods to the prediction of South African municipal financial statement audit outcomes.

HMC is the preferred MCMC algorithm due to its ability to intelligently explore the posterior through the use of first-order gradient information of the target via Hamiltonian dynamics. It turns out that when a magnetic field is added to HMC to produce non-canonical Hamiltonian dynamics, the target can be explored even more efficiently. This enhancement of HMC with a magnetic field leads to the MHMC algorithm. The MHMC method is a key focus of this thesis, and we enhance it in the following ways:

- We employ a random mass matrix for the auxiliary momentum variable to mimic the behaviour of quantum particles to create the QIMHMC algorithm. We prove that this new algorithm converges to the correct stationary distribution. Furthermore, experiments across various target distributions show that QIMHMC outperforms HMC, MHMC and QIHMC on a time-normalised effective sample size basis.

A limitation of the proposed algorithm is the requirement for the user to manually specify the distribution of the mass matrix and tune the parameters of the distribution. As with MHMC, the magnetic component of QIMHMC still needs to be tuned. These are open areas of research.

- We utilise partial momentum refreshment, instead of a total refreshment, of the momentum variable to create PMHMC which enhances the performance of MHMC. The results show that this approach leads to significant improvements in performance over HMC, MHMC and PHMC on a time-normalised effective sample size basis. A drawback of this method is the need to tune the partial momentum refreshment parameter, which determines the extent of the momentum retention. Our analysis shows that higher parameter values are favourable, but these values tend to vary between different target distributions. Tuning this parameter is still an open area of research.

- We obtain the fourth-order modified Hamiltonian associated with the numerical integrator employed in MHMC to create the new SMHMC method, which employs partial momentum refreshment for the generation of the momentum. The results show that the new method outperforms MHMC and PMHMC on an effective sample size basis across numerous targets. This method has two main drawbacks. The first relates to having to tune the partial momentum refreshment parameter as with PMHMC. The second relates to the method's high execution time due to the computation of the non-separable Hamiltonian. The high execution time leads to a decrease in performance on a time-normalised basis. Reducing the execution time is a crucial area of improvement of the proposed algorithm.

Previous studies have shown that HMC's sampling performance decreases as the step size or the system size increases. Methods based on sampling from the shadow Hamiltonian instead of the true Hamiltonian have been shown to address this pathology. This is due to the shadow Hamiltonian being better conserved by the numerical integrator than the true Hamiltonian. One such method is the S2HMC algorithm, which utilises a processed leapfrog integrator to avoid the computationally expensive generation of the momentum. In this thesis, we extend S2HMC in the following ways:

- We employ partial momentum refreshment to enhance the performance of S2HMC. This was achieved by partially retaining the momentum used to generate the previous sample in the current momenta generation before passing these momenta to the processed leapfrog integrator. The results showed that this approach results in outperformance over S2HMC and PMHMC on an ESS basis. As with SMHMC, the drawback of this method is the requirement to tune the partial momentum refreshment parameter as well as the associated long execution times typical of shadow Hamiltonian methods.

- We propose a method for tuning the parameters of S2HMC which is based on substituting the leapfrog integrator in NUTS with the processed integrator. This is a first-in-literature on methods for tuning parameters of algorithms based on shadow Hamiltonians. This removes the need for the user to manually specify and tune the trajectory length and step sizes parameters. Tuning these parameters would otherwise require time-consuming pilot runs. The results show that the method produces better ESSs than NUTS, outperforms S2HMC on a time-normalised ESS basis but underperforms NUTS on a normalised ESS basis due to the long execution time of the proposed algorithm.

MCMC estimators have a higher variance when compared to classical Monte Carlo estimators due to auto-correlations present between the generated samples. In this thesis, we use antithetic sampling to tackle the high variance problem in HMC based methods. Antithetic sampling involves running two chains concurrently, with one of the chains having a negated momentum variable compared to the other chain. In this thesis, we present the antithetic versions of MHMC and S2HMC. The results show that the antithetic versions of the algorithms produce higher time-normalised effective sample sizes when compared to antithetic HMC and the non-antithetic versions of the algorithms. These results show the usefulness of incorporating antithetic sampling into samplers that employ Hamiltonian dynamics. As this approach is straightforward and requires minimal assumptions about the target, there seems very little reason not to deploy it in practice.

We further provide a first-in-literature application of Bayesian inference to the analysis of municipal financial statement audit outcomes to understand the financial state

of South African local government entities. Within this Bayesian framework, we can extract the most important features for modeling audit outcomes through the use of ARD. We trained a BLR model with ARD on this dataset using the random walk MH, MALA, HMC, MHMC and S2HMC methods. The analysis shows that the repairs and maintenance as a percentage of total assets ratio, current ratio, debt to total operating revenue, net operating surplus margin, and capital cost to total operating expenditure ratio are the important features when predicting local government audit outcomes using financial ratios. These results could be useful for auditors as focusing on these ratios can speed up the detection of fraudulent behaviour in municipal entities and improve the speed and quality of the overall audit.

10.2 Ongoing and Future Work

The analysis performed in this thesis relied on the minimal tuning of the magnetic field in MHMC. We plan to address this drawback by assessing various methods for automatically tuning the magnetic field in future work. The manual tuning of the magnetic component performed in this thesis suggests that if the magnetic field is optimally tuned, it could significantly improve the performance of the MHMC algorithm. We are also currently investigating possible heuristics and automated approaches to automatically tune the partial momentum refreshment parameter in the PMHMC and PS2HMC methods. The QIMHMC algorithm can also be refined by developing automated approaches to select an optimal distribution from a user-specified class of distributions.

The slow execution times associated with evaluating shadow Hamiltonians in S2HMC and SMHMC can be addressed through the use of surrogate model approaches. A surrogate approach that could be considered is the use of Sparse grid [171] interpolation to pre-compute the gradients of the Hamiltonian before the sampling process begins. This would improve the time-normalised performance of the methods, but possibly at the cost of accuracy. Furthermore, the shadow Hamiltonian itself could be learned during the burn-in period using Gaussian Processes. As evaluating the shadow Hamiltonian is the most expensive part of the algorithms, this should reduce the execution time post-burn-in period and consequently improve the time-normalised effective sample size performance.

Furthermore, when we developed SMHMC, we only focused on a shadow Hamiltonian that is only conserved up to fourth-order. The results could be improved by considering higher-order shadow Hamiltonians, albeit at a higher computation cost.

One of the key issues in machine learning is model selection. Within the Bayesian framework, model selection and comparison are conducted via the Bayesian evidence metric. We are currently extending the MHMC methods presented in this thesis so that they are also able to produce the evidence metric. The approach that we are currently considering is creating a continuously tempered version of MHMC in a similar fashion to continuously tempered HMC outlined in [57]. Given than MHMC outperforms HMC across various targets, one would expect continuously tempered MHMC to outperform continuously tempered HMC.

The application areas of the methods presented in this thesis could be extended to larger datasets such as MNIST and deeper neural networks as opposed to the small neural networks considered in this thesis. This will consequently result in longer execution times. This could be improved by considering stochastic gradient versions of the algorithms proposed in this work. Currently, the literature does not cover stochastic gradient approaches for shadow Hamiltonian based samplers. Addressing this gap in the literature could prove to be a significant contribution to knowledge.

Lastly, in this thesis, we only considered variance reduction techniques based on using antithetic sampling. In the future, we plan to create a control variate version of MHMC. As control variate techniques are more generally applicable compared to antithetic sampling (which assumes symmetry of the target distribution), we expect control variate MHMC and the combination of control variate and antithetic sampling MHMC to outperform the antithetic MHMC method presented in this thesis. Furthermore, comparing the antithetic methods introduced in this thesis with the antithetic versions of RMHMC and QIHMC could be of interest.

Bibliography

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[2] H. M. Afshar, R. Oliveira, and S. Cripps. Non-volume preserving hamiltonian monte carlo and no-u-turnsamplers. In *International Conference on Artificial Intelligence and Statistics*, pages 1675–1683. PMLR, 2021.

[3] Y. Aït-Sahalia, C. Li, and C. X. Li. Closed-form implied volatility surfaces for stochastic volatility models with jumps. *Journal of Econometrics*, 222(1):364–392, 2021.

[4] E. Akhmatskaya and S. Reich. The targeted shadowing hybrid monte carlo (tshmc) method. In *New Algorithms for Macromolecular Simulation*, pages 145–158. Springer-Verlag, 2006.

[5] A. Alaa and M. Van Der Schaar. Frequentist uncertainty in recurrent neural networks via blockwise influence functions. In *International Conference on Machine Learning*, pages 175–190. PMLR, 2020.

[6] M. Alghalith. Pricing options under simultaneous stochastic volatility and jumps: A simple closed-form formula without numerical/computational methods. *Physica A: Statistical Mechanics and its Applications*, 540:123100, 2020.

[7] I. Amara, A. B. Amar, and A. Jarboui. Detection of fraud in financial statements:

French companies as a case study. *International Journal of Academic Research in Accounting, Finance and Management Sciences*, 3(3):40–51, 2013.

[8] C. Andrieu and J. Thoms. A tutorial on adaptive mcmc. *Statistics and computing*, 18(4):343–373, 2008.

[9] Y. F. Atchadé. An adaptive version for the metropolis adjusted langevin algorithm with a truncated drift. *Methodology and Computing in applied Probability*, 8(2):235–254, 2006.

[10] Auditor-General South Africa. Background to the three aspects we audit. https://www.agsa.co.za/portals/0/AGSA_Terminology.pdf, 2011. Last accessed: 16 August 2020.

[11] Auditor-General South Africa. MFMA 2018 - 2019. https://www.agsa.co.za/Reporting/MFMAReports/MFMA2018-2019.aspx, Jul 2020. Last accessed: 16 August 2020.

[12] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021.

[13] B. Bai, J. Yen, and X. Yang. False financial statements: Characteristics of China's listed companies and cart detecting approach. *International Journal of Information Technology & Decision Making*, 7(02):339–359, 2008.

[14] N. Bakhvalov. The optimization of methods of solving boundary value problems with a boundary layer. *USSR Computational Mathematics and Mathematical Physics*, 9(4):139–166, 1969.

[15] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12(4):273–288, 2000.

[16] M. Betancourt. Generalizing the no-u-turn sampler to riemannian manifolds. *arXiv preprint arXiv:1304.1920*, 2013.

[17] M. Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[18] M. Betancourt and M. Girolami. Hamiltonian monte carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79(30):2–4, 2015.

[19] L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[20] N. Bou-Rabee, A. Eberle, R. Zimmer, et al. Coupling and convergence for hamiltonian monte carlo. *Annals of Applied Probability*, 30(3):1209–1250, 2020.

[21] S. L. Boumediene. Detection and prediction of managerial fraud in the financial statements of Tunisian banks. *Accounting & Taxation*, 6(2):1–10, 2014.

[22] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[23] W. W. Bratton. Enron and the dark side of shareholder value. *Tul. L. Rev.*, 76:1275, 2001.

[24] F.-X. Briol, C. J. Oates, M. Girolami, and M. A. Osborne. Frank-wolfe bayesian quadrature: Probabilistic integration with theoretical guarantees. *arXiv preprint arXiv:1506.02681*, 2015.

[25] J. A. Brofos and R. R. Lederman. Magnetic manifold hamiltonian monte carlo. *arXiv preprint arXiv:2010.07753*, 2020.

[26] J. A. Brofos and R. R. Lederman. Non-canonical hamiltonian monte carlo. *arXiv preprint arXiv:2008.08191*, 2020.

[27] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[28] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

[29] C. M. Campos and J. M. Sanz-Serna. Extra chance generalized hybrid monte carlo. *Journal of Computational Physics*, 281:365–374, 2015.

[30] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: a probabilistic programming language. *Grantee Submission*, 76(1):1–32, 2017.

[31] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, January 2014.

[32] K.-C. Chen. Noncanonical poisson brackets for elastic and micromorphic solids. *International Journal of Solids and Structures*, 44(24):7715–7730, 2007.

[33] S. Chen. Detection of fraudulent financial statements using the hybrid data mining approach. *SpringerPlus*, 5(1):1–16, January 2016.

[34] Y.-J. Chen. On fraud detection method for narrative annual reports. In *The Fourth International Conference on Informatics & Applications (ICIA2015)*, pages 121–129, 2015.

[35] A. D. Cobb, A. G. Baydin, A. Markham, and S. J. Roberts. Introducing an explicit symplectic integration scheme for riemannian manifold hamiltonian monte carlo. *arXiv preprint arXiv:1910.06243*, 2019.

[36] R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223 – 236, 2001.

[37] J. Cotterill. Steinhoff shareholders sue Deloitte for damages. https://www.ft.com/content/4f4d591a-6f0f-11e8-92d3-6c13e5c92914, June 2018. Last accessed: 15 November 2019.

[38] J. Cronje. Steinhoff's market cap a mere R20bn as shares drop another 30%. https://www.fin24.com/Companies/Retail/steinhoff-shares-drop-by-a-fifth-in-early-trade-20171220, Dec 2017. Last accessed: 15 November 2019.

[39] I. K. Şen and S. Terzi. Detecting falsified financial statements using data mining: Empirical research on finance sector in turkey. *Maliye Finans Yazilari*, 26(96):67–82, 2012.

[40] L. Donnelly. Pic still limping from Steinhoff blow. https://mg.co.za/article/2018-06-08-00-pic-still-limping-from-steinhoff-blow, June 2018. Last accessed: 18 August 2020.

[41] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[42] S. Duane and J. B. Kogut. The theory of hybrid stochastic algorithms. *Nuclear Physics B*, 275(3):398–420, 1986.

[43] M. W. Dusenberry, D. Tran, E. Choi, J. Kemp, J. Nixon, G. Jerfel, K. Heller, and A. M. Dai. Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 204–213, 2020.

[44] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):1–9, 2021.

[45] K. M. Fanning and K. O. Cogger. Neural network detection of management fraud using published financial data. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 7(1):21–41, March 1998.

[46] M. A. Fernández-Gámez, F. García-Lagos, and J. R. Sánchez-Serrano. Integrating corporate governance and financial variables for the identification of qualified audit opinions with neural networks. *Neural Computing and Applications*, 27(5):1427–1444, June 2015.

[47] G. S. Fishman and B. D. Huang. Antithetic variates revisited. *Communications of the ACM*, 26(11):964–971, 1983.

[48] A. Frigessi, J. Gasemyr, and H. Rue. Antithetic coupling of two gibbs sampler chains. *Annals of Statistics*, pages 1128–1149, 2000.

[49] C. Gaganis. Classification techniques for the identification of falsified financial statements: a comparative analysis. *Intelligent Systems in Accounting, Finance & Management*, 16(3):207–229, July 2009.

[50] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[51] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[52] S. Ghosh, P. Birrell, and D. De Angelis. Variational inference for nonlinear ordinary differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 2719–2727. PMLR, 2021.

[53] M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[54] F. H. Glancy and S. B. Yadav. A computational model for financial reporting fraud detection. *Decision Support Systems*, 50(3):595–601, 2011.

[55] P. W. Glynn and C.-h. Rhee. Exact estimation for markov chain equilibrium expectations. *Journal of Applied Probability*, 51(A):377–389, 2014.

[56] Google Finance. Google Finance. https://www.google.com/finance/. Last accessed: 15 August 2021.

[57] M. Graham and A. Storkey. Continuously tempered hamiltonian monte carlo. In *Conference on Uncertainty in Artificial Intelligence*, 2017.

[58] Z. Grundiene. The model of fraud detection in financial statements by means of financial ratios. *Procedia-Social and Behavioral Sciences*, 213:321–327, 2015.

[59] M. Gu and S. Sun. Neural langevin dynamical sampling. *IEEE Access*, 8:31595–31605, 2020.

[60] T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S. J. Roberts. Sampling for inference in probabilistic models with fast bayesian quadrature. In *Advances in neural information processing systems*, pages 2789–2797, 2014.

[61] R. Gupta and N. S. Gill. Financial statement fraud detection using text mining. *International Journal of Advanced Computer Science and Applications*, 3(12), 2012.

[62] H. Haario, E. Saksman, and J. Tamminen. Adaptive proposal distribution for random walk metropolis algorithm. *Computational Statistics*, 14(3):375–395, 1999.

[63] E. Hairer. Backward error analysis for multistep methods. *Numerische Mathematik*, 84(2):199–232, 1999.

[64] E. Hairer, M. Hochbruck, A. Iserles, and C. Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

[65] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrica*, 57(1):97 –109, 1970.

[66] M. Haugh. Mcmc and bayesian modeling. http://www.columbia.edu/~mh2078/MachineLearningORFE/MCMC_Bayes.pdf, 2017. Last accessed: 15 September 2021.

[67] P. M. Healy and K. G. Palepu. The fall of Enron. *Journal of Economic Perspectives*, 17(2):3–26, May 2003.

[68] C. Heide, F. Roosta, L. Hodgkinson, and D. Kroese. Shadow manifold hamiltonian monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 1477–1485. PMLR, 2021.

[69] J. Heng and P. E. Jacob. Unbiased hamiltonian monte carlo with couplings. *Biometrika*, 106(2):287–302, 2019.

[70] G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

[71] M. Hoffman, A. Radul, and P. Sountsov. An adaptive-mcmc scheme for setting trajectory lengths in hamiltonian monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 3907–3915. PMLR, 2021.

[72] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

[73] M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[74] A. M. Horowitz. A generalized guided monte carlo algorithm. *Physics Letters B*, 268(2):247–252, 1991.

[75] A. Horowitz. Stochastic quantization in phase space. *Physics Letters B*, 156(1-2):89–92, 1985.

[76] S. L. Humpherys, K. C. Moffitt, M. B. Burns, J. K. Burgoon, and W. F. Felix. Identification of fraudulent financial statements using linguistic credibility analysis. *Decision Support Systems*, 50(3):585–594, February 2011.

[77] J. A. Izaguirre and S. S. Hampton. Shadow hybrid monte carlo: An efficient propagator in phase space of macromolecules. *J. Comput. Phys.*, 200(2):581–604, November 2004.

[78] P. E. Jacob, J. O'Leary, and Y. F. Atchadé. Unbiased markov chain monte carlo with couplings. *arXiv preprint arXiv:1708.03625*, 2017.

[79] V. E. Johnson. Studying convergence of markov chain monte carlo algorithms using coupled sample paths. *Journal of the American Statistical Association*, 91(433):154–166, 1996.

[80] V. E. Johnson. A coupling-regeneration scheme for diagnosing convergence in markov chain monte carlo algorithms. *Journal of the American Statistical Association*, 93(441):238–248, 1998.

[81] T. R. Kiehl, B. K. Hoogs, C. A. LaComb, and D. Senturk. Evolving multi-variate time-series patterns for the discrimination of fraudulent financial filings. In *Proc. of Genetic and Evolutionary Computation Conference*. Citeseer, 2005.

[82] E. Kirkos, C. Spathis, and Y. Manolopoulos. Data mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*, 32(4):995–1003, May 2007.

[83] L. Kish. *Survey sampling*. Number 04. HN29, K5., 1965.

[84] A. Klimke and B. Wohlmuth. Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in matlab. *ACM Transactions on Mathematical Software (TOMS)*, 31(4):561–579, 2005.

[85] T. Kohonen and T. Honkela. Kohonen network. *Scholarpedia*, 2(1):1568, 2007.

[86] S. Kotsiantis, E. Koumanakos, D. Tzelepis, and V. Tampakas. Financial application of neural networks: two case studies in Greece. In *Artificial Neural Networks – ICANN 2006*, pages 672–681. Springer Berlin Heidelberg, 2006.

[87] A. S. Kwekha-Rashid, H. N. Abduljabbar, and B. Alhayani. Coronavirus disease (covid-19) cases analysis using machine-learning applications. *Applied Nanoscience*, pages 1–13, 2021.

[88] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6405–6416, 2017.

[89] D. Levy, M. D. Hoffman, and J. Sohl-Dickstein. Generalizing hamiltonian monte carlo with neural networks. *arXiv preprint arXiv:1711.09268*, 2017.

[90] J. W. Lin, M. I. Hwang, and J. D. Becker. A fuzzy neural network for assessing the risk of fraudulent financial reporting. *Managerial Auditing Journal*, 18(8):657–665, November 2003.

[91] Z. Liu and Z. Zhang. Quantum-inspired hamiltonian monte carlo for bayesian sampling. *arXiv preprint arXiv:1912.01937*, 2019.

[92] M. López-Marcos, J. Sanz-Serna, and R. D. Skeel. Explicit symplectic integrators using hessian–vector products. *SIAM Journal on Scientific Computing*, 18(1):223–238, 1997.

[93] J. Ma, V. Rokhlin, and S. Wandzura. Generalized gaussian quadrature rules for systems of arbitrary functions. *SIAM Journal on Numerical Analysis*, 33(3):971–996, 1996.

[94] J. R. Macey. Efficient capital markets, corporate disclosure, and Enron. *Cornell Law Review*, 89:394–422, 2004.

[95] D. J. MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

[96] D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[97] R. Mbuvha, I. Boulkaibet, and T. Marwala. Bayesian automatic relevance determination for feature selection in credit default modelling. In *International Conference on Artificial Neural Networks*, pages 420–425. Springer, 2019.

[98] R. Mbuvha and T. Marwala. Bayesian inference of covid-19 spreading rates in south africa. *PloS one*, 15(8):e0237126, 2020.

[99] R. Mbuvha, W. T. Mongwe, and T. Marwala. Separable shadow hamiltonian hybrid monte carlo for bayesian neural network inference in wind speed forecasting. *Energy and AI*, page 100108, 2021.

[100] U. K. Mertens, A. Voss, and S. Radev. Abrox—a user-friendly python module for approximate bayesian computation with a focus on model comparison. *PloS one*, 13(3), 2018.

[101] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1–2):125 – 144, 1976.

[102] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, USA, 1995.

[103] P. R. Miles. pymcmcstat: A python package for bayesian inference using delayed rejection adaptive metropolis. *Journal of Open Source Software*, 4(38):1417, 2019.

[104] A. Mobiny, P. Yuan, S. K. Moulik, N. Garg, C. C. Wu, and H. Van Nguyen. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports*, 11(1):1–14, 2021.

[105] S. O. Moepya, S. S. Akhoury, and F. V. Nelwamondo. Applying cost-sensitive classification for financial fraud detection under high class-imbalance. In *2014 IEEE International Conference on Data Mining Workshop*, pages 183–192. IEEE, December 2014.

[106] S. O. Moepya, S. S. Akhoury, F. V. Nelwamondo, and B. Twala. The role of imputation in detecting fraudulent financial reporting. *International Journal of Innovative Computing, Information and Control*, 12(1):333–356, 2016.

[107] W. T. Mongwe and K. M. Malan. The efficacy of financial ratios for fraud detection using self organising maps. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1100–1106. IEEE, 2020.

[108] W. T. Mongwe. Analysis of equity and interest rate returns in south africa under the context of jump diffusion processes. Masters Thesis at University of Cape Town https://open.uct.ac.za/bitstream/handle/11427/16600/thesis_com_2015_mongwe_wilson_tsakane.pdf?isAllowed=y&sequence=1, Dec 2015. Last accessed: 18 March 2021.

[109] W. T. Mongwe and K. M. Malan. A survey of automated financial statement fraud detection with relevance to the south african context. *South African Computer Journal*, 32(1), 2020.

[110] W. T. Mongwe, R. Mbuvha, and T. Marwala. Adaptive magnetic hamiltonian monte carlo. *IEEE Access*, 9:152993–153003, 2021.

[111] W. T. Mongwe, R. Mbuvha, and T. Marwala. Adaptively setting the path length for separable shadow hamiltonian hybrid monte carlo. *IEEE Access*, 9:138598–138607, 2021.

[112] W. T. Mongwe, R. Mbuvha, and T. Marwala. Antithetic magnetic and shadow hamiltonian monte carlo. *IEEE Access*, 9:49857–49867, 2021.

[113] W. T. Mongwe, R. Mbuvha, and T. Marwala. Antithetic riemannian manifold and quantum-inspired hamiltonian monte carlo. *arXiv preprint arXiv:2107.02070*, 2021.

[114] W. T. Mongwe, R. Mbuvha, and T. Marwala. Bayesian inference of local government audit outcomes. *Plos one, https://10.1371/journal.pone.0261245*, 2021.

[115] W. T. Mongwe, R. Mbuvha, and T. Marwala. Locally scaled and stochastic volatility metropolis-hastings algorithms. *Algorithms*, 14(12), 2021.

[116] W. T. Mongwe, R. Mbuvha, and T. Marwala. Magnetic hamiltonian monte carlo with partial momentum refreshment. *IEEE Access*, 9:108009–108016, 2021.

[117] W. T. Mongwe, R. Mbuvha, and T. Marwala. Quantum-inspired magnetic hamiltonian monte carlo. *Plos one*, 16(10):e0258277, 2021.

[118] W. T. Mongwe, R. Mbuvha, and T. Marwala. Utilising partial momentum refreshment in separable shadow hamiltonian hybrid monte carlo. *IEEE Access*, 9:151235–151244, 2021.

[119] National Treasury Republic Of South Africa. MFMA circular no. 71. http://mfma.treasury.gov.za/Circulars/Pages/Circular71.aspx, 2014. Last accessed: 16 August 2020.

[120] National Treasury Republic Of South Africa. Municipal finance data. https://municipaldata.treasury.gov.za/, 2019. Last accessed: 16 August 2020.

[121] R. M. Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems 5*, pages 475–482, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[122] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, University of Toronto, Department of Computer Science, 01 1993.

[123] R. M. Neal. Slice sampling. *Annals of statistics*, 31(3):705–767, 2003.

[124] R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[125] R. M. Neal. Circularly-coupled markov chain sampling. *arXiv preprint arXiv:1711.04399*, 2017.

[126] R. M. Neal. Non-reversibly updating a uniform [0, 1] value for metropolis accept/reject decisions. *arXiv preprint arXiv:2001.11950*, 2020.

[127] R. M. Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

[128] S. H. Nia. Financial ratios between fraudulent and non-fraudulent firms: Evidence from Tehran stock exchange. *Journal of Accounting and Taxation*, 7(3):38–44, 2015.

[129] J. O'Donnel. Who's to blame for Wirecard? Germany passes the buck. https://www.reuters.com/article/us-wirecard-accounts-responsibility/whos-to-blame-for-wirecard-germany-passes-the-buck-idUSKBN24320O, Jul 2020. Last accessed: 18 August 2020.

[130] P. Omid, N. pour Hossein, and A. Zeinab. Identifying qualified audit opinions by artificial neural networks. *African Journal of Business Management*, 6(44):11077–11087, November 2012.

[131] M. Osborne, R. Garnett, Z. Ghahramani, D. K. Duvenaud, S. J. Roberts, and C. E. Rasmussen. Active learning of model evidence using bayesian quadrature. In *Advances in Neural Information Processing Systems*, pages 46–54, 2012.

[132] J. Owens and A. Hunter. Application of the self-organising map to trajectory classification. In *Proceedings Third IEEE International Workshop on Visual Surveillance*, pages 77–83. IEEE, 2000.

[133] J. Perols. Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, 30(2):19–50, 2011.

[134] J. L. Perols and B. A. Lougee. The relation between earnings management and financial statement fraud. *Advances in Accounting*, 27(1):39–53, 2011.

[135] O. S. Persons. Using financial statement data to identify factors associated with fraudulent financial reporting. *Journal of Applied Business Research (JABR)*, 11(3):38, 2011.

[136] D. Piponi, M. D. Hoffman, and P. Sountsov. Hamiltonian monte carlo swindles. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020.

[137] C. Presence. SA pensioners unlikely to recoup losses from Steinhoff. https://www.iol.co.za/business-report/companies/sa-pensioners-unlikely-to-recoup-losses-from-steinhoff-former-cfo-16795506, August 2018. Last accessed: 15 November 2019.

[138] S. J. Press. A compound events model for security prices. *Journal of business*, 40(3):317–335, 1967.

[139] PricewaterhouseCoopers. Global economic crime and fraud survey 2018 – South Africa. https://www.pwc.co.za/en/assets/pdf/gecs-2018.pdf, Feb 2018. Last accessed: 15 November 2019.

[140] T. Radivojević and E. Akhmatskaya. Mix & match hamiltonian monte carlo. *arXiv preprint arXiv:1706.04032*, 2017.

[141] Q. I. Rahman and G. Schmeisser. Characterization of the speed of convergence of the trapezoidal rule. *Numerische Mathematik*, 57(1):123–138, 1990.

[142] T. V. N. Rao, A. Gaddam, M. Kurni, and K. Saritha. Reliance on artificial intelligence, machine learning and deep learning in the era of industry 4.0. *Smart Healthcare System Design: Security and Privacy Aspects*, page 281, 2021.

[143] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge, R. Prudden, A. Mandhane, A. Clark, A. Brock, K. Simonyan, R. Hadsell, N. Robinson, E. Clancy, A. Arribas, and S. Mohamed. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, September 2021.

[144] C. Robert and G. Casella. *A short history of MCMC: Subjective recollections from incomplete data*. Chapman and Hall/CRC, 2011.

[145] G. O. Roberts, J. S. Rosenthal, et al. Optimal scaling for various metropolis-hastings algorithms. *Statistical science*, 16(4):351–367, 2001.

[146] J. S. Rosenthal. Faithful couplings of markov chains: now equals forever. *Advances in Applied Mathematics*, 18(3):372–381, 1997.

[147] V. Roy. Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, 7:387–412, 2020.

[148] F. Ruiz and M. Titsias. A contrastive divergence for combining variational inference and mcmc. In *International Conference on Machine Learning*, pages 5537–5545. PMLR, 2019.

[149] T. Salimans, D. Kingma, and M. Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226. PMLR, 2015.

[150] H. M. Schilit. Financial shenanigans: Detecting accounting gimmicks that destroy investments (corrected november 2010). *CFA Institute Conference Proceedings Quarterly*, 27(4):67–74, December 2010.

[151] R. D. Skeel and D. J. Hardy. Practical construction of modified hamiltonians. *SIAM Journal on Scientific Computing*, 23(4):1172–1188, 2001.

[152] D. B. Skillicorn and L. Purda. Detecting fraud in financial reports. In *2012 European Intelligence and Security Informatics Conference*, pages 7–13. IEEE, 2012.

[153] J. Sohl-Dickstein, M. Mudigonda, and M. DeWeese. Hamiltonian monte carlo without detailed balance. In *International Conference on Machine Learning*, pages 719–726. PMLR, 2014.

[154] X.-P. Song, Z.-H. Hu, J.-G. Du, and Z.-H. Sheng. Application of machine learning methods to risk assessment of financial statement fraud: Evidence from China. *Journal of Forecasting*, 33(8):611–626, October 2014.

[155] C. R. Sweet, S. S. Hampton, R. D. Skeel, and J. A. Izaguirre. A separable shadow hamiltonian hybrid monte carlo method. *The Journal of chemical physics*, 131(17):174106, 2009.

[156] N. Tripuraneni, M. Rowland, Z. Ghahramani, and R. Turner. Magnetic hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 3453–3461. PMLR, 2017.

[157] R.-H. Tsaih, W.-Y. Lin, and S.-Y. Huang. Exploring fraudulent financial reporting with GHSOM. In *Intelligence and Security Informatics*, pages 31–41. Springer Berlin Heidelberg, 2009.

[158] A. W. Van der Stoep, L. A. Grzelak, and C. W. Oosterlee. The heston stochastic-local volatility model: efficient monte carlo simulation. *International Journal of Theoretical and Applied Finance*, 17(07):1450045, 2014.

[159] K. Vanslette, A. Al Alsheikh, and K. Youcef-Toumi. Why simple quadrature is just as good as monte carlo. *Monte Carlo Methods and Applications*, 26(1):1–16, 2020.

[160] D. Vats, J. M. Flegal, and G. L. Jones. Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*, 106(2):321–337, 04 2019.

[161] M. Veraar. The stochastic fubini theorem revisited. *Stochastics An International Journal of Probability and Stochastic Processes*, 84(4):543–551, 2012.

[162] Y. Wang and D. M. Blei. Frequentist consistency of variational bayes. *Journal of the American Statistical Association*, 114(527):1147–1161, 2019.

[163] Z. Wang and N. de Freitas. Predictive adaptation of hybrid monte carlo with bayesian parametric bandits. In *NIPS Deep Learning and Unsupervised Feature Learning Workshop*, volume 30, 2011.

[164] Z. Wang, S. Mohamed, and N. Freitas. Adaptive hamiltonian and riemann manifold monte carlo. In *International conference on machine learning*, pages 1462–1470. PMLR, 2013.

[165] R. Wehrens and L. M. C. Buydens. Self- and super-organizing maps in R: The kohonen package. *Journal of Statistical Software*, 21(5):1–19, 2007.

[166] R. Wehrens and J. Kruisselbrink. Flexible self-organizing maps in kohonen 3.0. *Journal of Statistical Software*, 87(7):1–18, 2018.

[167] K. Xu, H. Ge, W. Tebbutt, M. Tarek, M. Trapp, and Z. Ghahramani. Advancedhmc. jl: A robust, modular and efficient implementation of advanced hmc algorithms. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–10. PMLR, 2020.

[168] J. Yang, G. O. Roberts, and J. S. Rosenthal. Optimal scaling of random-walk metropolis algorithms on general target distributions. *Stochastic Processes and their Applications*, 130(10):6094–6132, 2020.

[169] I.-C. Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

[170] J. H. Zar. Spearman rank correlation. *Encyclopedia of biostatistics*, 7, 2005.

[171] C. Zenger and W. Hackbusch. Sparse grids. In *Proceedings of the Research Workshop of the Israel Science Foundation on Multiscale Phenomenon, Modelling and Computation*, page 86, 1991.

[172] C. Zhang, B. Shahbaba, and H. Zhao. Precomputing strategy for hamiltonian monte carlo method based on regularity in parameter space. *Computational Statistics*, 32(1):253–279, 2017.

Appendix A

Summary of Audit Outcome Literature Survey

In this appendix, we present a summary of the findings from the FSF detection and audit outcomes literature survey that we provide in our article in [109]. This summary uses the themes of the most common: 1) definitions of fraud, 2) features used for detecting fraud, 3) region of the case study, dataset size and imbalance, 4) algorithms used for detection, 5) approach to feature selection / feature engineering, 6) treatment of missing data, and 7) performance measure used in the literature.

Tables A.1 to A.3 provides a summary of some of the aspects of FSF detection as reflected in the 52 papers surveyed in [109], with a visual representation of changes over time. Table A.1 shows three different definitions of FSF and the total number of studies from the survey that have used each definition. In the table, each digit '1' under a time period indicates a study using the given definition of fraud. Similarly, Tables A.2 and A.3 show the data feature type and the most commonly used detection methods, respectively. In Table A.3, the digit '2' is used to indicate two studies using the given method. Note that most studies used multiple methods, so the totals do not correspond to the number of studies for each period. Table A.4 summarises the overall findings from the survey, where the percentages in the brackets show the proportion of studies in the survey that used the particular approach.

Table A.1: FSF definitions used through time. Each digit 1 in the table represents a study that used a given definition of FSF in the time period, where A: Investigation by authorities, Q: Qualified audit opinion, C: Combination of both definitions.

Definition	Total	1995–1999	2000–2004	2005–2009	2010–2014	2015–2018
A	33 (63%)	111	1	111111	111111111111	11111111111
Q	12 (23%)			1111	11111	111
C	7 (13%)	1	1	111	11	
Totals	52	4	2	13	19	14

Table A.2: FSF data features used through time. Each digit 1 in the table represents a study that used a given type of data feature in the time period, where F: financial ratios, F&NF: financial and non-financial ratios, T: text, and F&T: financial variables and text.

Data feature	Total	1995–1999	2000–2004	2005–2009	2010–2014	2015–2018
F	27 (52%)	111	11	1111111111	11111111	1111
F&NF	16 (31%)	1		111	11111111	1111
T	7 (13%)				111	1111
F&T	2 (4%)					11
Totals	52	4	2	13	19	14

Table A.3: The FSF detection methods through time. Each digit 2 in the table represents two studies that used the given detection method in the time period, where LR: logistic regression, ANN: artificial neural network, SVM: support vector machine, DT: decision tree, DA: discriminant analysis, and OTH: other.

Method	Total	1995–1999	2000–2004	2005–2009	2010–2014	2015–2018
LR	24 (18%)	2	2	222	22222	22
ANN	26 (21%)	2		222	22222	2222
SVM	16 (13%)			2	222	2222
DT	14 (12%)			22	222	22
DA	6 (6%)	2		22		
OTH	36 (29%)			22222	2222222222	222
Totals	122	6	2	32	52	30

Table A.4: Summary of findings from the FSF detection literature survey

Implementation issue	**Most common approach in the literature**
Fraud definition	Investigations by authorities (63%)
Data features	Financial ratios (52%)
Data imbalance	Match fraud firms with non-fraud firms (71%)
Data region	USA (38%) and Taiwan (13%)
Data size	min (27), mean (2 365), median (190), max (49 039)
Methods used	ANN (21%), logistic regression (18%) and SVM (13%)
Feature selection	Filter based approaches (69%)
Missing data treatment	Not specified or delete records (94%)
Performance measures	Classification accuracy (35%)
Learning approach	Supervised classification (97%)
Best FSF detection method	Varies across datasets

Appendix B

Derivation of Separable Shadow Hamiltonian

Using the BCH [64] formula, it can be shown that the fourth-order shadow Hamiltonian is given as:

$$\tilde{\mathrm{H}}^{[4]}(\mathbf{w},\mathbf{p}) = \mathrm{H}(\mathbf{w},\mathbf{p}) + \frac{\epsilon^2}{12}\mathrm{K}_{\mathbf{p}}^{T}\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} - \frac{\epsilon^2}{24}\mathrm{U}_{\mathbf{w}}^{T}\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^4) \tag{B.1}$$

One can interpret the first error term in equation (B.1) as a perturbation to the kinetic energy K and the next as a perturbation to the potential energy U. The perturbation to the kinetic energy K alters the fixed mass matrix $\mathbf{M}$ into a different and likely position-dependent matrix. In most situations, this makes the distribution of the momentum variable non-Gaussian.

We now seek a canonical transformation that can transform the Hamiltonian in Equation (B.1) to a separable Hamiltonian. Sweet *et al.* [165] use the following canonical transformation [92]:

$$\begin{aligned} \mathbf{W} &= \mathbf{w} + \epsilon^2\lambda\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^2) \\ \mathbf{P} &= \mathbf{p} - \epsilon^2\lambda\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} + \mathcal{O}(\epsilon^2) \end{aligned} \tag{B.2}$$

After substitution of (B.2) into (B.1) the shadow Hamiltonian becomes:

$$\begin{aligned}\tilde{\mathrm{H}}^{[4]}(\mathcal{X}(\mathbf{w},\mathbf{p})) &= \mathrm{H}(\mathbf{W},\mathbf{P}) + \frac{\epsilon^2}{12}\mathrm{K}_{\mathbf{P}}\mathrm{U}_{\mathbf{WW}}\mathrm{K}_{\mathbf{P}} - \frac{\epsilon^2}{24}\mathrm{U}_{\mathbf{W}}\mathrm{K}_{\mathbf{PP}}\mathrm{U}_{\mathbf{W}} + \mathcal{O}(\epsilon^4) \\ &= \underbrace{\mathrm{U}(\mathbf{W})}_{\mathrm{A}} + \underbrace{\mathrm{K}(\mathbf{P})}_{\mathrm{B}} + \underbrace{\frac{\epsilon^2}{12}\mathrm{K}_{\mathbf{P}}\mathrm{U}_{\mathbf{WW}}\mathrm{K}_{\mathbf{P}}}_{\mathrm{C}} - \underbrace{\frac{\epsilon^2}{24}\mathrm{U}_{\mathbf{W}}\mathrm{K}_{\mathbf{PP}}\mathrm{U}_{\mathbf{W}}}_{\mathrm{D}} + \mathcal{O}(\epsilon^4)\end{aligned} \tag{B.3}$$

where taking the first order Taylor expansion of $\mathrm{U}(\mathbf{W})$ at $\mathbf{w}$ and substituting $\mathbf{W}$ gives:

$$\begin{aligned}A &= \mathrm{U}(\mathbf{w}) + \mathrm{U}_{\mathbf{w}}(\mathbf{W}-\mathbf{w}) + \frac{1}{2}(\mathbf{W}-\mathbf{w})\,\mathrm{U}_{\mathbf{ww}}(\mathbf{W}-\mathbf{w})^T + \mathcal{O}(\epsilon^4) \\ &= \mathrm{U}(\mathbf{w}) + \mathrm{U}_{\mathbf{w}}\left(\epsilon^2\lambda\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}}\right) + \left(\epsilon^2\lambda\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}}\right)\mathrm{U}_{\mathbf{ww}}\left(\epsilon^2\lambda\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}}\right)^T + \mathcal{O}(\epsilon^4) \\ &= \mathrm{U}(\mathbf{w}) + \mathrm{U}_{\mathbf{w}}\epsilon^2\lambda\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^4)\end{aligned} \tag{B.4}$$

and substituting $\mathbf{P}$ into B gives:

$$\begin{aligned}B &= \frac{1}{2}\left(\mathbf{p} - \epsilon^2\lambda\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} + \mathcal{O}(\epsilon^2)\right)\mathrm{K}_{\mathbf{pp}}\left(\mathbf{p} - \epsilon^2\lambda\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} + \mathcal{O}(\epsilon^2)\right) \\ &= \mathrm{K}(\mathbf{p}) - \epsilon^2\lambda\mathrm{K}_{\mathbf{p}}\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} + \mathcal{O}(\epsilon^4)\end{aligned} \tag{B.5}$$

By using the second order Taylor approximation of $\mathrm{U}(\mathbf{W})$ at $\mathbf{w}$ and substituting $\mathbf{W}$, we have that:

$$\begin{aligned}\mathrm{U}(\mathbf{W}) &= \mathrm{U}(\mathbf{w}) + \mathrm{U}_{\mathbf{w}}(\mathbf{W}-\mathbf{w}) + \frac{1}{2}(\mathbf{W}-\mathbf{w})\,\mathrm{U}_{\mathbf{ww}}(\mathbf{W}-\mathbf{w})^T + \mathcal{O}(\epsilon^4) \\ \mathrm{U}_{\mathbf{W}} &= \mathrm{U}_{\mathbf{w}} + \mathrm{U}_{\mathbf{ww}}(\mathbf{W}-\mathbf{w}) + \mathcal{O}(\epsilon^4) \\ \mathrm{U}_{\mathbf{WW}} &= \mathrm{U}_{\mathbf{ww}} + \mathcal{O}(\epsilon^4)\end{aligned} \tag{B.6}$$

We then have that:

$$\begin{aligned}C &= \frac{\epsilon^2}{12}\mathrm{K}_{\mathbf{P}}\mathrm{U}_{\mathbf{WW}}\mathrm{K}_{\mathbf{P}} \\ &= \frac{\epsilon^2}{12}\mathrm{K}_{\mathbf{P}}\left(\mathrm{U}_{\mathbf{ww}} + \mathcal{O}(\epsilon^4)\right)\mathrm{K}_{\mathbf{P}} \\ &= \frac{\epsilon^2}{12}\mathrm{K}_{\mathbf{P}}\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{P}} + \mathcal{O}(\epsilon^4)\end{aligned} \tag{B.7}$$

Similarly, from the first order Taylor approximation of $\mathrm{U}(\mathbf{W})$ at $\mathbf{w}$, it follows that

$\mathrm{U}_{\mathbf{W}} = \mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^2)$ and noting that $\mathrm{K}_{\mathbf{PP}} = \mathrm{K}_{\mathbf{pp}}$ as K is quadratic, we have that:

$$\begin{aligned} D &= \frac{\epsilon^2}{24}\mathrm{U}_{\mathbf{W}}\mathrm{K}_{\mathbf{PP}}\mathrm{U}_{\mathbf{W}} \\ &= \frac{\epsilon^2}{24}\left(\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^2)\right)\mathrm{K}_{\mathbf{pp}}\left(\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^2)\right) \\ &= \frac{\epsilon^2}{24}\mathrm{U}_{\mathbf{w}}\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^4) \end{aligned} \tag{B.8}$$

It then follows that:

$$\begin{aligned} \tilde{\mathrm{H}}^{[4]}(\mathcal{X}(\mathbf{w},\mathbf{p})) &= A + B + C + D \\ &= \mathrm{H}(\mathbf{w},\mathbf{p}) + \epsilon^2\left(\frac{1}{12} - \lambda\right)\mathrm{K}_{\mathbf{p}}\mathrm{U}_{\mathbf{ww}}\mathrm{K}_{\mathbf{p}} + \epsilon^2\left(\lambda - \frac{1}{24}\right)\mathrm{U}_{\mathbf{w}}\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^4) \end{aligned} \tag{B.9}$$

To remove the second term which contains mixed derivatives of $\mathbf{w}$ and $\mathbf{p}$ and noting that $\mathrm{K}_{\mathbf{pp}}$ is a constant, we set $\lambda = \frac{1}{12}$ and have that:

$$\tilde{\mathrm{H}}^{[4]}(\mathcal{X}(\mathbf{w},\mathbf{p})) = \mathrm{H}(\mathbf{w},\mathbf{p}) + \frac{\epsilon^2}{24}\mathrm{U}_{\mathbf{w}}\mathrm{K}_{\mathbf{pp}}\mathrm{U}_{\mathbf{w}} + \mathcal{O}(\epsilon^4) \tag{B.10}$$

which is the required separable shadow Hamiltonian which is used in S2HMC.

Appendix C

S2HMC Satisfies Detailed Balance

Theorem C.0.1. *S2HMC satisfies detailed balance.*

Proof. This proof is taken from Sweet *et al.* [155] Section III A. To show that S2HMC satisfies detailed balance, we need to show that the processed leapfrog integration scheme is both sympletic and reversible. That is, we need to show that the processing map $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \mathbf{p})$ commutes with reversal of momenta and preserves phase space volume so that the resulting processed leapfrog integrator used in S2HMC is both sympletic and reversible and thus ensures detailed balance.

We can get a symplectic map using a generating function of the third kind [92]:

$$S(\mathbf{w}, \hat{\mathbf{p}}) = \mathbf{w}\hat{\mathbf{p}} + \frac{\epsilon}{24}\big[\mathrm{U}_{\mathbf{w}}(\mathbf{w} + \epsilon\mathbf{M}^{-1}\hat{\mathbf{p}}) - \mathrm{U}_{\mathbf{w}}(\mathbf{w} - \epsilon\mathbf{M}^{-1}\hat{\mathbf{p}})\big] \tag{C.1}$$

The map $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \mathbf{p})$ is then given by:

$$\hat{\mathbf{w}} = \frac{\partial S}{\partial \hat{\mathbf{p}}}; \quad \mathbf{p} = \frac{\partial S}{\partial \mathbf{w}}. \tag{C.2}$$

The map $(\hat{\mathbf{w}}, \hat{\mathbf{p}}) = \mathcal{X}(\mathbf{w}, \mathbf{p})$ from equation (C.1) is given precisely by equations (2.33). This map is the pre-processing step of S2HMC. In other words, it is the canonical change in variables that preserves the symplectic property of the processed leapfrog algorithm. The inverse mapping $(\mathbf{w}, \mathbf{p}) = \mathcal{X}^{-1}(\hat{\mathbf{w}}, \hat{\mathbf{p}})$ is given by equation (2.34). The inverse mapping is the post-processing step of S2HMC. The reversibility of the processed leapfrog algorithm can be shown by $\mathcal{X}(\mathbf{w}, -\mathbf{p}) = \mathrm{diag}(I, -I)\mathcal{X}(\mathbf{w}, \mathbf{p})$. Thus, since the processed leapfrog algorithm is both symplectic and reversible, S2HMC preserves detailed balance. □

Appendix D

Derivatives From Non-Canonical Poisson Brackets

In this appendix, we present the derivatives derived from the non-canonical Poisson brackets:

$$\begin{aligned} \{K, U\} &= -\nabla_{\mathbf{p}} K \nabla_{\mathbf{w}} U + \nabla_{\mathbf{w}} K \nabla_{\mathbf{p}} U + \nabla_{\mathbf{p}} K \mathbf{G} \nabla_{\mathbf{p}} U - K_{\mathbf{p}} U_{\mathbf{w}} \\ \{K, \{K, U\}\} &= -K_{\mathbf{p}} U_{\mathbf{ww}} K_{\mathbf{p}} - K_{\mathbf{p}} \mathbf{G} K_{\mathbf{pp}} U_{\mathbf{w}} \\ \{U, K\} &= U_{\mathbf{w}} K_{\mathbf{p}} \\ \{U, \{U, K\}\} &= U_{\mathbf{w}} K_{\mathbf{pp}} U_{\mathbf{w}} \end{aligned} \tag{D.1}$$

where $\nabla_a f = \frac{\partial f}{\partial a}$ for some function f which is a function of some variable a. Note that the derivatives from the non-canonical Poisson brackets differ from the canonical derivatives through the presence of the magnetic field $\mathbf{G}$. When $\mathbf{G} = \mathbf{0}$, the non-canonical Poisson bracket derivatives collapse to the canonical derivatives.