University of Southampton

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Adaptive Market Making using Reinforcement Learning in a Multi-Agent Market Simulation

DOI: 10.5258/SOTON/PG/T316

by

Christopher Jaehoon Cho
MA (Cantab)
ORCiD: 0000-0003-2805-0352

A thesis for the degree of
Doctor of Philosophy

11 May 2026

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

Adaptive Market Making using Reinforcement Learning in a Multi-Agent Market Simulation

by Christopher Jaehoon Cho

Financial markets are amongst the most complex and well-studied multi-agent systems. Over the past several decades, researchers have devoted considerable effort to understanding their structural and behavioural properties. However, building a simulated market environment that can replicate realistic interactive behaviour remains an ongoing challenge. This limitation restricts the scope and validity of experiments aimed at testing new trading strategies or hypotheses, particularly when simplifying assumptions, such as negligible market impact or the absence of market feedback, fail to hold.

This thesis addresses this challenge by building on the Agent-Based Interactive Discrete Event Simulation (ABIDES) simulation platform to create an empirically grounded and reproducible environment for studying cryptocurrency markets. First, we develop a methodology for tuning simulation parameters to replicate the stylised facts observed in Binance following the 2017 cryptocurrency boom. Second, we introduce Price-Reverting Impact Model of a cryptocurrency Exchange (PRIME), a novel configuration of the ABIDES simulator designed to produce realistic market responses to agent actions whilst retaining the ability to track an external price series. Third, we use the PRIME framework to conduct the first controlled comparison of Reinforcement Learning (RL) based market makers across various architectures in a realistic, interactive environment, establishing a benchmark for future research.

Our findings demonstrate that a carefully calibrated agent population comprising Zero Intelligence, Momentum, and Mean-Reversion agents can reproduce key statistical properties observed in the Bitcoin market. Furthermore, the PRIME framework successfully captures market impact behaviours described in existing and novel empirical observations, allowing for controlled but realistic experimentation of market agents in this environment. Finally, our reinforcement learning market-makers deployed on this platform reveal important insights into the trade-offs involved in state-action space design and the learning dynamics of various RL architectures. This setup also allows for a meaningful comparative evaluation of market-makers with previous work. Collectively, these contributions provide a reproducible foundation for advancing market simulation frameworks and for developing and evaluating novel reinforcement learning approaches to the market-making problem.

Contents

List of Figures **ix**

List of Tables **xiii**

Declaration of Authorship **xix**

Acknowledgements **xxi**

1 Introduction **1**
1.1 Research Challenges . . . 3
1.1.1 Availability of data . . . 3
1.1.2 Heterogeneity of participants . . . 3
1.1.3 Non-stationarity of the Financial Markets . . . 4
1.2 Research objectives . . . 4
1.3 Research contributions . . . 6
1.4 Structure of the thesis . . . 8

2 Literature Review **11**
2.1 Continuous Double Auction (CDA) . . . 11
2.1.1 Experimental Economics . . . 13
2.1.2 Modelling the exchange . . . 14
2.1.2.1 Theoretical equilibrium model . . . 15
2.1.2.2 Stochastic approach . . . 15
2.1.2.3 Statistical methods . . . 16
2.1.2.4 Machine learning models . . . 17
2.1.2.5 Agent based models . . . 17
2.1.3 Stylised Facts . . . 17
2.1.4 Market impact . . . 18
2.1.5 Summary . . . 19
2.2 Agent Based Modelling (ABM) . . . 20
2.2.1 Agent Categorisation . . . 23
2.2.2 Market Making agents . . . 24
2.2.2.1 Optimal control based traders . . . 24
2.2.2.2 Reinforcement learning based traders . . . 25
2.2.2.3 Online learning based traders . . . 28
2.2.3 Fundamental agents . . . 28
2.2.3.1 Zero Intelligence . . . 29
2.2.3.2 Assignment-Adaptive . . . 30

2.2.3.3 GD (Gjerstad and Dickhaut) . . . 30
2.2.3.4 Adaptive-Aggressiveness . . . 31
2.2.4 Technical agents . . . 32
2.2.4.1 Momentum . . . 32
2.2.4.2 Mean Reversion . . . 33
2.2.4.3 Noise traders . . . 33
2.2.5 Environment . . . 33
2.2.5.1 Project PLATO . . . 34
2.2.5.2 Santa Fe Artificial Stock Market . . . 34
2.2.5.3 GEM/MAGENTA . . . 35
2.2.5.4 Bristol Stock Exchange (BSE) . . . 36
2.2.5.5 ABIDES . . . 36
2.2.5.6 ExPo . . . 37
2.2.6 Summary . . . 37
2.3 Reinforcement Learning . . . 37
2.3.1 History of Reinforcement Learning . . . 38
2.3.2 Types of learning . . . 40
2.3.2.1 Temporal Differencing vs Monte-Carlo . . . 40
2.3.2.2 Model-free vs Model-based . . . 40
2.3.2.3 Types of Model-free Reinforcement learning . . . 41
2.3.2.4 On-policy vs Off-policy . . . 42
2.3.3 Function approximators . . . 43
2.3.3.1 DQN . . . 44
2.3.3.2 PPO . . . 45
2.3.4 Summary . . . 46
2.4 Bitcoin Market Microstructure . . . 46
2.4.1 Stylised facts . . . 47
2.4.2 Market impact . . . 48
2.4.3 Summary . . . 49

3 Simulating the Bitcoin exchange 51
3.1 Market Participants . . . 51
3.1.1 Zero-Intelligence . . . 52
3.1.2 Zero-Intelligence Plus . . . 52
3.1.3 Momentum . . . 56
3.1.4 Mean-Reversion . . . 57
3.1.5 Market-Maker . . . 57
3.2 Desired stylised facts . . . 58
3.2.1 Asset returns . . . 58
3.2.2 Market volume . . . 59
3.3 Cryptocurrency exchange . . . 59
3.3.1 Data collection . . . 60
3.3.2 Properties of the market data . . . 61
3.4 Tuning the simulation . . . 62
3.4.1 Fixing trade volume . . . 63
3.4.2 Initial results . . . 64
3.4.3 Manual tuning . . . 64

3.4.4 Manual tuning results . . . 67
3.4.5 Macro parameter tuning . . . 68
3.4.6 Macro tuning results . . . 70
3.4.7 Response surface method . . . 71
3.4.8 Tuned simulation result . . . 72
3.4.9 Tuned simulation setup . . . 73
3.4.10 Computational considerations . . . 75
3.5 Conclusion . . . 77

4 Incorporating Market Impact **79**
4.1 Defining Market Impact . . . 79
4.1.1 Shape of the Limit Order Book . . . 80
4.1.2 Temporal Market Impact . . . 80
4.2 Bitcoin (BTC)/United States Dollar (USD) Futures . . . 81
4.2.1 Initial Impact . . . 81
4.2.2 Impact Reversion . . . 85
4.2.3 Meta Orders . . . 86
4.3 Baseline Simulation . . . 88
4.3.1 Initial Impact . . . 88
4.4 Zero-Intelligence simulation . . . 90
4.5 Santa-Fe Model . . . 90
4.5.1 Initial Impact . . . 91
4.5.2 Impact Reversion . . . 92
4.5.3 Meta Orders . . . 94
4.5.4 Meta Order Augmentation . . . 95
4.5.5 Model evaluation . . . 98
4.6 PRIME . . . 99
4.6.1 Price Tracking . . . 101
4.6.2 Selective Liquidity Taking . . . 103
4.6.3 Initial Impact . . . 105
4.6.4 Impact Reversion . . . 107
4.6.5 Meta Orders . . . 110
4.6.6 Adding technical agents . . . 111
4.6.6.1 Initial Impact . . . 111
4.6.6.2 Impact Reversion . . . 113
4.6.6.3 Meta Orders . . . 115
4.6.7 Simulation configuration . . . 116
4.6.8 Model evaluation . . . 118
4.7 Conclusion . . . 119

5 Benchmarking Market-Makers **121**
5.1 Recap of market making . . . 122
5.2 Recap of reinforcement learning . . . 123
5.3 Market-Maker Design . . . 124
5.3.1 State Space Design . . . 124
5.3.2 Action Space Design . . . 126
5.3.3 Reward Function Design . . . 126

5.4 Simulation Setup . . . 128
5.4.1 Market-making agents . . . 128
5.4.2 Benchmark model: Spooner *et al.* (2018) . . . 129
5.4.3 Simulation Length . . . 131
5.4.4 Hyperparameters and Experimental Settings . . . 131
5.5 Simulation Results . . . 132
5.5.1 Varying the state-action space design . . . 134
5.5.1.1 Compact State-Action Space Simulation . . . 134
5.5.1.2 Extended State-Space Simulation . . . 137
5.5.1.3 Adjusted Action-Space Simulation . . . 140
5.5.1.4 Full State-Action Space Simulation . . . 143
5.5.1.5 Evaluation of training results . . . 146
5.5.2 Evaluation of trained models . . . 148
5.5.2.1 Empirical results . . . 148
5.5.3 Discussion . . . 150
5.6 Conclusion . . . 151

6 Conclusions and Future Work **155**
6.1 Research Outcome . . . 155
6.2 Future Work . . . 157

References **159**

List of Figures

3.1 Memory usage comparison between ZIP and GDX 56
3.2 BTC/USD stylised facts (March 2021) 62
3.3 BTC/USD stylised facts vs Untuned simulation results 1 65
3.4 BTC/USD stylised facts vs Untuned simulation results 2 66
3.5 BTC/USD stylised facts vs micro-parameter tuning results 1 69
3.6 BTC/USD stylised facts vs micro-parameter tuning results 2 70
3.7 Simulated vs Actual midprice for varying number of technical agents . . 71
3.8 BTC/USD stylised facts vs Final tuned results 1 73
3.9 BTC/USD stylised facts vs Final tuned results 2 74

4.1 Average BTC/USD LOB Volume (Actual Data, September 2021) 81
4.2 BTC/USD Market Impact (5 sec resample) 82
4.3 BTC/USD Market Impact (5 sec resample, volume & volatility adjusted) 83
4.4 BTC/USD Market Impact with percentile bucket average impact (5 sec resample, volume & volatility adjusted) 83
4.5 BTC/USD order percentile bucket average impact (5 sec resample, volume & volatility adjusted) 84
4.6 BTC/USD order percentile bucket average impact (5 sec resample, volume & volatility adjusted, including exponent) 84
4.7 BTC/USD Market Impact (5 sec resample, volume & volatility adjusted with exponent) 85
4.8 BTC/USD order percentile bucket average impact (5 sec resample, volume & volatility adjusted, including exponent) 86
4.9 BTC/USD market impact coefficient over time (5 sec resample, volume & volatility adjusted, including exponent) 87
4.10 BTC/USD market impact decay over time (5 sec resample, volume & volatility adjusted, including exponent) 87
4.11 BTC/USD Autocorrelation of Order Sign (Actual Data, March 2021) . . 88
4.12 Baseline Market Impact (5 sec resample) 89
4.13 Baseline Market Impact with percentile bucket average impact (5 sec resample, volume & volatility adjusted) 89
4.14 Zero-Intelligence BTC/USD Market Impact with percentile bucket average impact (1 sec resample) 90
4.15 Santa-Fe Market Impact with percentile bucket average impact (5 sec resample) 92
4.16 Zero-Intelligence BTC/USD Market Impact with percentile bucket average impact (1 sec resample) 93

4.17 Santa-Fe Market Impact with percentile bucket average impact by previous order direction (5 sec resample) . . . 93
4.18 Santa-Fe market impact coefficient over time (5 sec resample) . . . 94
4.19 Santa-Fe market impact decay over time (5 sec resample) . . . 94
4.20 Autocorrelation of Order Sign (Santa-Fe Simulation) . . . 95
4.21 Autocorrelation of Order Sign (Simulated Data, March 2021) . . . 96
4.22 Grid search for the parameters of DAR(p) process . . . 97
4.23 Autocorrelation of Order Sign (Simulated Data, March 2021) . . . 98
4.24 Price paths (5 second snapshot, varying observation error, July 2020) . . 101
4.25 Price deviation (5 second snapshot, varying observation error, July 2020) 102
4.26 Price deviation (5-second snapshot, 1000 limit order agents, July 2020) . 102
4.27 Price deviation (5 second snapshot, 1000 limit order agents, July 2020) . 103
4.28 Autocorrelation of Order Sign (Simulated Data, March 2021) . . . 104
4.29 Autocorrelation of Order Sign (Simulated Data, March 2021) . . . 104
4.30 Autocorrelation of Order Sign (Simulated Data, July 2020) . . . 105
4.31 PRIME Market Impact (5 sec resample) . . . 105
4.32 PRIME net order volume histogram (5 sec resample) . . . 106
4.33 PRIME net order volume histogram (5 sec resample) . . . 106
4.34 PRIME Market Impact (5 sec resample, adjusted) . . . 107
4.35 PRIME Market Impact (5 sec resample, adjusted) . . . 107
4.36 PRIME market impact coefficient over time (5 sec resample) . . . 108
4.37 PRIME market impact coefficient over time (5 sec resample) . . . 109
4.38 PRIME market impact coefficient over time (5 sec resample) . . . 109
4.39 PRIME market impact coefficient over time (10 sec resample) . . . 109
4.40 PRIME market impact coefficient over time (10 sec resample) . . . 110
4.41 PRIME autocorrelation of Order Sign (Simulated Data, March 2021) . . 110
4.42 PRIME Market Impact (5 sec resample, adjusted) . . . 112
4.43 PRIME Market Impact Comparison (5 sec resample, adjusted) . . . 112
4.44 Histogram of total order volume for middle 5% of net trade volume in PRIME simulation (5 sec resample) . . . 113
4.45 Histogram of price change for middle 5% of net trade volume in PRIME simulation (5 sec resample) . . . 113
4.46 PRIME Market Impact (5 sec resample, adjusted) . . . 114
4.47 PRIME market impact coefficient over time (5 sec resample) . . . 114
4.48 PRIME market impact coefficient over time (5 sec resample) . . . 115
4.49 PRIME market impact decay over time (5 sec resample) . . . 115
4.50 PRIME autocorrelation of Order Sign (Simulated Data, July 2020) . . . 116
4.51 PRIME autocorrelation of Order Sign (Simulated Data, July 2020) . . . 117

5.1 Profit and loss (PnL) per volume over time (compact state-action space setup with 4 observable states and 5 discrete actions) . . . 135
5.2 Inventory levels over time (compact state-action space setup with 4 observable states and 5 discrete actions) . . . 135
5.3 Average quoted spread (compact state-action space setup with 4 observable states and 5 discrete actions) . . . 136
5.4 Average volume traded per wake-up over time (compact state-action space setup with 4 observable states and 5 discrete actions) . . . 136

5.5 Reward over time (compact state-action space setup with 4 observable states and 5 discrete actions) . . . 137
5.6 Inventory levels over time (extended state-space setup with 8 observable states and 5 discrete actions) . . . 138
5.7 Average quoted spread (extended state-space setup with 8 observable states and 5 discrete actions) . . . 138
5.8 Average volume traded per wake-up over time (extended state-space setup with 8 observable states and 5 discrete actions) . . . 139
5.9 PnL per volume over time (extended state-space setup with 8 observable states and 5 discrete actions) . . . 139
5.10 Reward over time (extended state-space setup with 8 observable states and 5 discrete actions) . . . 140
5.11 Inventory levels over time (adjusted action-space setup with 8 observable states and 6 discrete actions) . . . 141
5.12 Average quoted spread (adjusted action-space setup with 8 observable states and 6 discrete actions) . . . 141
5.13 Average volume traded per wake-up over time (adjusted action-space setup with 8 observable states and 6 discrete actions) . . . 142
5.14 PnL per volume over time (adjusted action-space setup with 8 observable states and 6 discrete actions) . . . 142
5.15 Reward over time (adjusted action-space setup with 8 observable states and 6 discrete actions) . . . 143
5.16 Inventory levels over time (full state-action space with 8 observable states and 10 discrete actions) . . . 144
5.17 Average quoted spread (full state-action space with 8 observable states and 10 discrete actions) . . . 145
5.18 Average volume traded per wake-up over time (full state-action space with 8 observable states and 10 discrete actions) . . . 145
5.19 PnL per volume over time (full state-action space with 8 observable states and 10 discrete actions) . . . 146
5.20 Reward over time (full state-action space with 8 observable states and 10 discrete actions) . . . 146
5.21 Distribution of mean inventory held by each agent across 1000 12-minute market sessions (plot shows median, middle 50% and middle 90% of data) 149
5.22 Distribution of final PnL per trade across 1000 12-minute market sessions (plot shows median, middle 50% and middle 90% of data) . . . 149
5.23 Distribution of mean spreads by each agent across 1000 12-minute market sessions (plot shows median, middle 50% and middle 90% of data) . . . 150

List of Tables

3.1 Tuned simulation configuration 76

4.1 PRIME Simulation Agent Configuration 118

5.1 Simulation environment configuration (PRIME framework) 132
5.2 Training protocol and episode design 132
5.3 DQN hyperparameters used in Chapter 5 experiments 132
5.4 PPO hyperparameters used in Chapter 5 experiments 132
5.5 A2C hyperparameters used in Chapter 5 experiments 133

Glossary

limit order An order to buy or sell a set quantity at a specified price or better, which may not execute immediately. 12, 13, 16, 19, 22, 26, 31, 52, 59, 68, 80, 81, 90, 91, 93, 94, 98, 99, 102, 104, 111, 117, 124, 126, 134, 140, 145, 147

mark-to-market Profit and loss computed by valuing open positions using current market prices rather than execution prices. 127, 136, 139, 147

market impact The change in price caused by executing a trade, typically increasing with order size or aggressiveness. iii, 5–8, 11, 14, 18, 19, 21, 22, 28, 37, 40, 46, 48, 49, 61, 79–86, 88–90, 92, 95, 97–99, 105, 107–109, 111, 113, 114, 119–121, 123, 153, 155–157

market order An order to buy or sell a set quantity immediately at the best available price. 12, 26, 32, 48, 49, 59, 68, 81, 91, 95, 98–104, 110, 111, 116, 117, 119, 124, 127, 130, 134, 140, 153

mid-price The average of the best bid and ask prices; often used as an estimate of the true market value. 18, 22, 57, 60, 80, 82, 91, 99–103, 110, 116, 117, 126, 127, 134, 137, 140

slippage The difference between the expected trade price and the actual executed price, often due to market movement or impact. 5, 6, 8, 18–21, 48, 87

spread The difference between the ask and bid prices. 5, 6, 8, 12, 20–22, 24, 27, 28, 48, 57, 121, 122, 134, 135, 137–139, 141, 144, 146, 147, 150–153

state-action space The full set of combinations of environment states and agent actions used in reinforcement learning. iii, 5, 6, 8, 26, 39, 41–43, 121–124, 129, 134, 146, 148, 151, 152, 155–157

stylised facts Empirical regularities consistently observed in financial markets, such as heavy tails, volatility clustering, and absence of autocorrelation in returns. iii, 3–6, 11, 17–19, 35, 46, 47, 49, 51, 58, 60, 61, 63, 64, 68, 71–73, 76–79, 89, 121

tick The smallest allowable price increment between different bids or offers on an exchange. 16, 20, 57, 74, 126, 134, 135, 140, 141, 144

Acronyms

A2C Advantage Actor-Critic. 8, 121, 128, 129, 134–136, 138, 140–142, 144, 145, 148, 150–153, 156

AA Adaptive-Aggressiveness. 31, 32, 36

ABIDES Agent-Based Interactive Discrete Event Simulation. iii, 4, 5, 7, 36, 37, 51, 56, 73, 75–77, 117, 119, 128, 155

ABM Agent-Based Model. 20–23

ASAD Assignment-Adaptive. 30

BTC Bitcoin. vii, 4, 5, 7, 51, 60, 61, 64, 73, 74, 78, 80–83, 85–88, 98, 105, 117, 119–121, 155

CARA Constant Absolute Risk Aversion. 26

CDA Continuous Double Auction. 1, 12–17, 19–21, 23, 28, 29, 34–37

CME Chicago Mercantile Exchange. 12, 13, 48

DQN Deep Q-Network. 8, 27, 44, 45, 121, 128, 129, 134–142, 144–148, 150, 152, 153, 156

FBA Frequent Batch Auction. 20, 21

GD Gjerstad and Dickhaut. 30, 31, 35

GDX Gjerstad Dickhaut eXtended. 31, 32, 36, 55, 56, 75

HFT High-Frequency Trader. 21, 27

HPC high-performance computing. 75

LOB Limit Order Book. 1–3, 12, 14, 16–19, 27, 36, 38, 43, 48, 52, 60, 80, 81, 88, 90, 99, 103, 111, 112, 124, 131, 151

LSE London Stock Exchange. 1, 22

MC Monte Carlo. 40

MDP Markov Decision Process. 38, 122, 123

ML Machine-Learning. 17, 122, 152

PnL Profit and loss. x, xi, 5, 8, 121, 130, 132, 135, 136, 139, 141, 142, 145, 146, 149, 151, 153

PPO Proximal Policy Optimisation. 8, 45, 121, 128, 129, 134–136, 138–148, 150, 152, 153, 156

PRIME Price-Reverting Impact Model of a cryptocurrency Exchange. iii, 7, 8, 79, 99–103, 105–108, 111, 115–121, 123, 128, 131, 148, 149, 151, 152, 156

RL Reinforcement Learning. iii, 2, 4–6, 8, 25–28, 37–40, 43–46, 121–124, 126, 128, 131, 132, 143, 148, 151–153, 155–157

SARSA State Action Reward State Action. 25, 41, 42

TD Temporal-Difference. 38–40

TWAP Time Weighted Average Price. 87, 119

USD United States Dollar. vii, 4, 5, 7, 51, 60, 61, 64, 73, 74, 78, 80–83, 85–88, 98, 105, 119–121, 155

VWAP Volume Weighted Average Price. 87, 119

ZI Zero-Intelligence. 29, 30, 33, 52–54, 70, 74, 75, 77, 90, 91, 117

ZIP Zero-Intelligence Plus. 29–32, 35, 36, 52–56, 69, 70, 74, 75, 77, 119

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
    - Cho and Norman (2021). *Bit by bit: how to realistically simulate a crypto-exchange*. In *Proceedings of the Second ACM International Conference on AI in Finance*, pp. 1–9.
    - Cho et al. (2023). *PRIME: A Price-Reverting Impact Model of a cryptocurrency Exchange*. In *Cryptocurrency Research Conference, Monaco*.
    - Cho *et al.* (2023). *Set the standard: Benchmarking model-free reinforcement learning market-makers in a multi-agent market simulator*. In *Finance and Business Analytics Conference, Greece*.

Signed:..... Date: 11 May 2026

Acknowledgements

I would like to express my sincere gratitude to Professor Tim Norman. His guidance, expertise, and support throughout the research process have been invaluable.

I am also deeply grateful to Dr Manuel Nunes for his insight, thoughtful feedback, and steadfast support, both as a colleague and as a supervisor in producing this work.

I would also like to thank the staff, colleagues, and friends at the University of Southampton's Agents, Interaction and Complexity research group and the Centre for Doctoral Training in Next Generation Computational Modelling.

Lastly, I would especially like to thank all my friends and colleagues in the lab in Building 32 for fostering such a friendly and supportive research environment. The casual gatherings over coffee, board games over lunch, evenings in pubs, late-night takeaways, Thursday-night karaokes, and many other shared moments became some of the most memorable parts of my PhD. These experiences helped me grow both personally and academically, and they are memories that I will carry with me for the rest of my life. I will always be grateful for the friendship, encouragement, and sense of community that made this journey so special.

This work was supported by the EPSRC Centre for Doctoral Training in Next Generation Computational Modelling (EP/L015382/1). I also gratefully acknowledge the use of the IRIDIS High Performance Computing Facility and the associated support services at the University of Southampton in the completion of this research.

To my father, for his unwavering presence in my life,

and to my mother, for finding her way back to happiness

사랑해요

Chapter 1

Introduction

The financial markets are vital for the modern-day economy. They enable efficient capital allocation with minimal friction, helping individuals, businesses, and entire nations manage their risks, exposures, and intertemporal wealth effectively. Thanks to this highly connected international financial system, resources can flow from one part of the world to another based on demand and supply.

The financial markets are divided broadly into public and private markets, where transactions occur openly and behind closed doors, respectively. Due to the opaqueness of the private markets, most research into the modern financial system is conducted in the public domain, where rules and data are more transparent and are subject to constant regulatory scrutiny. The public market can be divided into primary and secondary markets. The primary market is focused on the creation of financial instruments, bringing together issuers and investors, whilst the secondary market predominantly revolves around the exchange of preexisting assets, allowing buyers and sellers to discover fair value for their assets.

One of the most prominent examples of a secondary market is the London Stock Exchange (LSE). This electronic marketplace facilitates billions of transactions daily, connecting various market participants. The LSE mainly operates using a Continuous Double Auction (CDA) mechanism, supported by a central Limit Order Book (LOB), which records all public buy and sell orders. At any given moment, the state of the auction can be broadly assessed by examining the order book, which provides information on market activity. Although additional mechanisms such as closing auctions and block trades have been introduced to boost liquidity and foster competition, the core of the operations of the LSE remains centred on the central LOB and the CDA.

A more recent example of a secondary market is Binance, one of the largest cryptocurrency exchanges in the world. Over the past decade, the rapid rise of digital assets such as Bitcoin has brought about significant shifts in financial markets, both in retail

and institutional investment space. Bitcoin, created in 2009, is one of the most widely traded cryptocurrencies and has evolved into a major asset class, with several billion dollars of this digital asset changing hands daily. Although originally conceived as a decentralised digital currency, Bitcoin has increasingly been adopted by institutional investors and incorporated into centralised exchanges like Binance. As a result, Bitcoin is now traded in much the same way as traditional financial assets. Its market behaviour is shaped by principles similar to those in established financial exchanges, utilising a centralised LOB, continuous trading, asset indexation, and enhanced liquidity through derivative instruments such as futures and options. These features, combined with comprehensive and freely available market data from Binance, make Bitcoin an ideal subject for analysis and research, especially in data-hungry domains.

Given the sheer size of the financial system and its numerous participants, any player who can use information from the auction to accurately forecast market sentiment, impending transactions, available liquidity, or a host of other variables places themselves in a position to gain a substantial edge over others. Consequently, many researchers in academia, private entities, and governmental organisations have a vested interest in modelling financial exchanges as accurately as possible. For academics, an accurate model opens new avenues of research into the market mechanism and improves their ability to explain financial markets empirically and intuitively. For the private sector, it allows firms to evaluate the consequences of their actions in the market ex ante transactions, which may translate into higher profits. For the public sector, such models can help enhance the efficiency of the financial system whilst ensuring fairness is maintained across the system as a whole. Given these incentives, it is very likely that most institutional investors and investment banks in the private sector have their own proprietary models of the market to simulate, hypothesise, and evaluate the financial system. However, due to the strategic sensitivity of these tools, many of the most successful and accurate models are likely to remain proprietary and are not available to the public.

The lack of effective open solutions for such a crucial topic presents a wide range of research opportunities. This is particularly true for the growing cryptocurrency markets, where the history of academic research is relatively brief. In this thesis, we focus first on producing a simulation framework that can reproduce the dynamics of a financial exchange, treating Bitcoin as a traditional asset class within the framework of a centralised exchange. We then use our simulation framework of the financial exchange to compare techniques in RL that are being adopted within the financial sector. By leveraging the free market data provided by Binance in our research, we are able to explore and evaluate market dynamics in a manner that is both robust and explainable. In the remainder of this introduction, we will discuss the specific challenges and objectives of our research in more detail.

1.1 Research Challenges

To develop an accurate model of a financial exchange, we must ensure that the simulation is accurate and robust. Modern computational tools greatly aid in enhancing the model of an exchange, but there remain a handful of hurdles that hamper progress within this field.

1.1.1 Availability of data

The availability of data has already limited most research into the private markets. However, despite being open to the public, the vast majority of public financial exchanges also charge a certain fee to access their data. This cost increases by orders of magnitude when trying to obtain historical data with a sufficient level of information for research. In order to make accurate comparisons of a simulated market to real markets, a large time series of level 3 LOB (or more granular) data is required. This data refers to a transaction-by-transaction report from the exchange, which specifies the size and price of all orders that are submitted and cancelled from the exchange. As purchasing this data is not an option for most academic researchers including us, we circumvent this constraint by sourcing data from open platforms such as a cryptocurrency exchange and open source sample datasets. As data from these sources may exhibit unexpected behaviour or contain undesirable properties versus the traditional financial instruments, we dedicate part of our research to verifying many of the stylised facts observed in this dataset by comparing the market microstructure that is observed against literature from traditional markets and past work into stylised facts observed in the domain of cryptocurrencies.

1.1.2 Heterogeneity of participants

In an ideal world, every market participant would be analysed and modelled into the simulation to yield the most realistic outcome. However, this is unrealistic for two reasons. Firstly, in order to analyse the behaviour of every participant, LOB data including trader ID is required. However, this data is usually available only to the exchanges, as it can reveal private information such as trading strategies. Secondly, there are limits in the number of heterogeneous agents that can be added into the simulation, given limited time and computational resources endowed to us. On top of this, including individually tuned agents leaves the simulation susceptible to overfitting, causing the environment to be less generalisable and robust when it comes to scenario testing using information that was not observed previously. As a compromise, we limit the number of heterogeneous agents in our simulation by choosing a few representative models, but

we introduce a degree of heterogeneity amongst these agents by adding stochasticity to their hyperparameters.

1.1.3 Non-stationarity of the Financial Markets

The exchange we create is optimised and evaluated against the stylised facts collected from the cryptocurrency exchange. Unfortunately, the evolving nature of the market means that these stylised facts, which we take as ground truth in our research, can change over time. Therefore as the market evolves, the realism of our simulation may gradually deteriorate, potentially leading to behaviour that is not in line with the latest stylised facts. However, as predicting future path of market microstructure is very difficult (so much so that it can be a research topic on its own), we accept that a discrepancy may arise at a future date. However, we assume that the data we collect on market microstructure is a good representation of the market for the purpose of this research. In addition, we will leave sufficient information on our methodologies to allow future researchers to re-tune their models, if and when the stylised facts evolve.

1.2 Research objectives

The objectives of this thesis fall under two broad themes: developing an empirically grounded multi-agent simulation framework of a cryptocurrency exchange, and using this environment to evaluate RL approaches to market making. The first three objectives correspond to the former theme, whilst the final objective relates to the latter:

Objective 1: Extract and analyse the stylised facts from the BTC/USD market.

- What statistical regularities characterise the BTC/USD market across returns, order flow, and volume?
- How do these stylised facts compare with those documented in traditional financial markets, as documented by Ballocchi et al. (1999) and Cont (2001)?

Objective 2: Develop a methodology for tuning a multi-agent simulation on the ABIDES platform.

- To what extent can an empirically tuned simulation reproduce the stylised facts observed in real markets?
- How effectively can domain knowledge be used to tune agent behaviours and improve model fidelity?

- How does the resulting simulation compare with existing multi-agent exchange models, such as Vyetrenko et al. (2021)?

Objective 3: Incorporate market impact dynamics into the simulation.

- What approaches can be used to model market impact and its reversion within a multi-agent system?
- How effective are these approaches at reproducing market impact observed in the real world and incorporating realistic slippage?
- Can these dynamics be incorporated whilst retaining the ability to track an external price series for scenario testing?

Objective 4: Design and evaluate RL-controlled market makers in the simulated exchange developed previously.

- How effective are different RL architectures at learning to make markets within a realistic, interactive simulation environment?
- How does variation in state-action space design influence learning behaviour and overall performance across architectures?
- How does the performance of different architectures vary across key market-making metrics such as inventory management, quoted spread, traded volume, and realised profit and loss (PnL)?
- To what extent do market-making models developed using static historical data translate to performance in an interactive, multi-agent market simulation?

The first theme of this thesis looks at the development of an empirically grounded multi-agent simulation framework of a cryptocurrency exchange. A realistic simulation requires a well-defined representation of the market it seeks to emulate. For this reason, we place significant emphasis on extracting and analysing the stylised facts of the BTC/USD market, which serve as empirical targets for the behaviour of the synthetic system. These stylised facts capture the statistical properties of asset returns, volatility dynamics, volume patterns and more, and provide a benchmark for evaluating the realism of a synthetic market environment.

To reproduce these properties, we adopt an agent-based modelling technique using the ABIDES framework. Agent-based models are particularly suitable for this task, as they allow the behaviour of heterogeneous market participants to be expressed in an intuitive, interpretable, and modular manner. Unlike models based on stochastic processes or generative approaches, the behaviour of an agent-based simulation can be traced

directly to the decisions of individual agents, enabling transparent evaluation of the causal relationships within the system. This transparency supports both explainability and empirical tuning, as the contribution of each agent type to the aggregate market behaviour can be both individually and systematically examined.

Building on this modelling philosophy, we develop a methodology to tune the simulation such that the resulting synthetic market reproduces key microstructural properties observed in the real world. This includes long-tailed return distributions, volatility clustering, intraday volume characteristics and others. Beyond passive reproduction of stylised facts, we further extend the simulation by incorporating realistic market impact dynamics, which is central to understanding execution costs (or slippage), liquidity provision, and agent−environment interaction. By modelling both impact and its reversion, the simulation becomes capable of representing the feedback effects that arise when agents trade in a live market, enabling more realistic evaluation of trading strategies.

The second theme of this thesis is the evaluation of RL approaches to market making within this empirically grounded simulation environment. Market makers provide liquidity to the market by simultaneously quoting buy and sell prices (bids and asks), and their performance whilst doing so depends on a range of factors such as the agent's inventory, quoted spread, and prevailing market dynamics. Existing studies often rely on historical data or simplified market assumptions, which limits comparability across the literature. In some cases, researchers ignore market impact by assuming that the market maker is a small part of the market or trading in a highly liquid environment. Such assumptions reduce the realism of the learning task and hinder the transferability of results to real-world settings.

By training agents within a dynamic multi-agent simulation where all participants are subject to the same stochastic environment and realistic slippage, we are able to provide a controlled and comparable evaluation framework for market-making strategies. This setting allows us to examine how different reinforcement-learning architectures learn to make markets, how their performance changes under different state-action space designs, and how they compare against established approaches when evaluated under identical conditions. In doing so, we aim to produce a consistent and reusable benchmark for future research into applying RL to market making.

1.3 Research contributions

The research presented in this thesis addresses the four objectives outlined in Section 1.2, each corresponding to a gap in the existing literature on cryptocurrency market microstructure, multi-agent exchange simulation, and reinforcement-learning-based market making. The contributions of this thesis are structured around these objectives,

highlighting both empirical and methodological advances, as well as the practical implications arising from our work.

Our first contribution covering the first research objective is an updated empirical characterisation of the BTC/USD market, based on high-quality limit order book data obtained from Binance (Chapter 3 & 4). The literature on cryptocurrency market microstructure has not kept pace with the rapid evolution of this market, and several canonical stylised facts found in traditional markets (Ballocchi et al. (1999); Cont (2001)), such as return distribution behaviour, volatility clustering, volume dynamics, and order-flow autocorrelation, had not been explored or revisited since the structural changes to the market following the 2017 cryptocurrency rally (Fink and Johann (2014)). In addition, we provide updated empirical evidence of market impact (Donier and Bonart (2015)), and the first evidence of its reversion, and autocorrelation properties of orders in the cryptocurrency world, showing where cryptocurrency markets align and deviate from established microstructural regularities found in other markets. In addition, these findings provide an empirical foundation that serves as a target for the simulation framework developed in subsequent chapters.

The second contribution is the development of a novel methodology for tuning a multi-agent simulation of a cryptocurrency exchange on the ABIDES platform. This covers all three of our research questions from the second research objective, with our approach of combining domain knowledge with empirical technique in the form of a response surface method to adjust the behaviours and the ratios of heterogeneous agents yielding a synthetic market that reproduces key stylised facts identified previously. To our knowledge, this represents the first application of response surface methodology for tuning a multi-agent financial exchange, and the first multi-agent simulation framework specifically designed and empirically calibrated for cryptocurrency markets, ahead of more recent works using multi-agent models such as Cong et al. (2024); Mansurov et al. (2024), to understand aspects of the crypto ecosystem. The resulting environment significantly improves existing solutions such as Vyetrenko et al. (2021), particularly with respect to the reproduction of trade volumes, which previous work struggled to reproduce. Importantly, this contribution reduces the field's reliance on costly and often fragmented historical market data. By providing an empirically grounded synthetic exchange environment, we reduce the barrier to entry for future researchers looking into cryptocurrency market microstructure.

Our third contribution, corresponding to the third research objective, is the development of the PRIME framework that is capable of reproducing empirically observed market impact, its reversion, and the autocorrelation of order-flow. To our knowledge, this is the first instance in which this was achieved by tweaking the ratio of technical agents into the model as opposed to introducing a stochastic process to achieve the same goal. At the same time, the model retained the ability to follow an externally specified price series for scenario analysis by making a portion of the agent participation exert pressure

on the price if the simulated price deviates from the external price. To our knowledge, this is the first multi-agent simulation of a cryptocurrency exchange that incorporates realistic market impact behaviour, as other work on market impact focused on identification and not replication (Barucci et al. (2023); Silantyev (2019)). This extension is essential for modelling execution costs (slippage) and agent-environment feedback effects. By embedding market impact directly into the simulation, we allow for interactive experimentation in which agents can both influence and be influenced by the market, increasing the explanatory power and realism of our simulated exchange for order book research.

The final contribution of this thesis is the development and comparison of a suite of market-making agents based on state-of-the-art RL algorithms within the empirically grounded PRIME environment. This contribution addresses all four of our research questions related to the fourth research objective by training and evaluating multiple RL architectures (Deep Q-Network (DQN), Proximal Policy Optimisation (PPO), Advantage Actor-Critic (A2C)) under consistent simulation environments and analysing their performance across varying state-action space designs. This constitutes the first like-for-like comparison of reinforcement-learning-based market makers within a realistic, interactive, empirically tuned exchange simulation. Previous studies, such as Gašperov et al. (2021) and Spooner et al. (2018), relied on static historical data and made simplifying assumptions that removed market impact, thus limiting the study's generalisability. Our framework ensures that all algorithms face the same endogenous market conditions, enabling fair and reproducible benchmarking across strategies and across metrics such as inventory, spread and PnL. In addition, we compare our agents against a representative model from previous literature in this domain, to provide a like-for-like comparison between this model and our work in an environment that exhibits realistic microstructural complexity. This framework to compare performances of various market-makers provides the benchmark for future researchers to use when comparing the performance of their RL based market-making agents.

1.4 Structure of the thesis

This thesis consists of six chapters. In the current chapter, we laid out the landscape of our research in the domain of market simulations and market-makers. In Chapter 2, we present a comprehensive literature survey on existing methods of modelling the market, empirical studies on the cryptocurrency market, as well as an overview of reinforcement learning techniques that can be applied to the market-making problem. In Chapter 3 and 4, we analyse the data we collect and create a multi-agent simulation that can reproduce key properties found in the data. In Chapter 5, we introduce the market-maker's problem and provide a comparison between existing reinforcement learning

architectures applied to market-making. Finally, the thesis is wrapped up in Chapter 6, where we provide an overview of our contributions and directions for future research.

Chapter 2

Literature Review

The underpinning research we build upon in this thesis spans artificial intelligence (specifically agent-based simulation and reinforcement learning) and computational finance. We first address the core mechanism that underpins the financial exchange, the Continuous Double Auction (CDA), its properties, and the means of modelling it. This is followed by an overview of existing agent-based models (ABM), which details the properties of potential market participants. Then, we provide information on the latest developments in reinforcement learning and their application within the financial setting. Lastly, we provide a summary of stylised facts and market impact related literature from the Bitcoin market.

2.1 Continuous Double Auction (CDA)

An auction is a mechanism through which buyers and sellers interact to exchange goods or services at prices determined by the participants' private valuations. Auctions are used across various sectors, including digital platforms, financial markets, and non-financial exchanges such as art, real estate, and government procurement, and play a critical role in facilitating both resource allocation and price discovery. Resource allocation refers to the efficient transfer of goods or services to those who value them most, while price discovery is the process through which a market-clearing price (price at which the quantity supplied equals the quantity demanded) is identified by matching buyers and sellers within the market.

Participants in auctions form valuations based on a combination of private and public information, which influences their pricing strategies. Public information includes widely available data such as macroeconomic reports, corporate earnings, or market announcements, while private information refers to knowledge that an individual can obtain behind closed doors, such as internal sentiment or confidential discussions. The

integration of these different information types shapes each participant's reservation price - the maximum price a buyer is willing to pay or the minimum price a seller is willing to accept. The continuous incorporation of new information by participants via auctions contributes to the price formation process, which is essential to maintaining efficient, liquid, and fair markets.

While there exist many forms of auctions with slightly varied objectives (Krishna offers good coverage from English auctions to double auctions in his textbook Krishna (2009)), the type of auction we focus on is the CDA, which is the dominant auction format in modern financial markets. In a CDA, buyers and sellers interact continuously, submitting orders to buy or sell a homogenous product. The participants have two methods of buying and selling from the market in the form of limit orders and market orders. A limit order is an order to buy or sell the product at a specific price or better. These orders are not executed immediately unless the specified price can already be executed in the market, and instead remain in the market until they are matched with a counterparty or cancelled. On the other hand, a market order is an order to buy or sell the product immediately at the best available price on the market. This means that the transaction will occur immediately, but the transaction price is determined based on the availability of existing orders that are available in the market. In a CDA, these orders are kept in a centralised record called the LOB. The LOB keeps a dynamic record of all outstanding buy (bids) and sell (asks) limit orders for the asset, and ranks them by price. The bid-ask spread (the difference between the most competitive bid price and the ask price) represents the market's current buy/sell price for any participants looking for a unit of immediate transaction. Therefore, when a new order is submitted to the exchange, it is either matched with an existing order, resulting in a transaction, or added to the book if no immediate match can be found. The LOB is a crucial component of modern financial markets, facilitating continuous price discovery by providing transparency into the market, whilst fostering liquidity and minimising transaction costs.

A point to note is that the mechanism for matching a market order with limit orders can vary between markets and exchanges. Although there exists many matching algorithms that are currently being used in CDAs, we will give a few common examples that are being used in the financial markets. The most commonly used mechanism is the First In First Out (FIFO) algorithm, which matches existing limit orders at the most competitive price, followed by time priority of these orders (the oldest order at the same price level is executed first). The S&P 500 E-mini futures on the Chicago Mercantile Exchange (CME) is an example of a CDA that utilises the FIFO algorithm. Another mechanism is the Pro Rata algorithm, where limit orders at the most competitive price are executed proportionately to the volume offered. For example, if there are three bids at the most competitive price, with trade size of 100, 150 and 250 respectively, a market order to sell 100 units will be distributed amongst these bids, with each bid receiving an execution quantity of 20, 30 and 50 units, respectively. The AUD/USD Futures market on the

CME is a good example of a market that is matched using the Pro Rata algorithm. The final example of matching algorithms deployed in a CDA is Size Priority. As opposed to being executed on a time priority as in the FIFO, Size Priority executes the largest limit orders at the most competitive price first. This encourages liquidity provision and reduces order fragmentation. An example of size-priority matching can be found in the corporate bond market on Tradeweb.

As discussed in Chapter 1, our project aims to create a simulation framework that can reproduce the dynamics of a cryptocurrency exchange, which can then be used to compare the performance of market-makers that are controlled via reinforcement learning. With this in mind, the next section provides a clear summary of CDA from an experimental economic perspective. From there, we provide a broad overview of the existing methods for modelling the CDA, followed by a detailed explanation of our chosen modelling approach.

2.1.1 Experimental Economics

The study of double auctions was first brought into light by the Nobel Prize-winning experimental economist Vernon Smith in 1962 (Smith (1962)). In his paper in the Journal of Political Economy, he conducted various experiments with his students to derive experimental insights into the properties of a CDA. He divided his students into groups of buyers and sellers, assigning each a private valuation of a fictitious good. Participants were incentivised to maximise their profits by optimising the difference between the transaction price and their private valuation: for sellers, this was calculated as the sale price minus their private valuation, and for buyers, it was their private valuation less the purchase price. The auction was further divided into "trading days", during which each participant was typically limited to a single transaction per session. In most cases, private valuations received by individuals were carried over to subsequent trading days.

Within this setting, Smith adjusted the private valuation assigned to each participant to control the demand and supply schedule of the market. Since he knew the private valuation of every participant, he was able to calculate the theoretical market equilibrium prior to running his tests. The results showed that after a couple of trading sessions, human traders posted results that were very close to the maximum allocative efficiency of the market - the extent to which resources are distributed in a way that maximises total utility, indicating that goods are allocated to the buyers who value them the most from the suppliers who can produce them at the lowest cost. In addition to this, Smith was able to provide experimental evidence for existing ideas such as the Walrasian hypothesis (Walras (2013)), which states that markets will naturally converge toward an equilibrium where supply equals demand, as a result of the iterative adjustment of prices by participants in response to excess demand or supply. The most striking takeaway from

Smith's experiments was the ability of the untrained participants to reach the competitive equilibrium in a CDA. Although there were differences in the rate of convergence when the experiments were carried out on graduate students reading Economics, tests carried out on novice candidates with little to no understanding of economic theory also converged towards the theoretical equilibrium. This experiment demonstrated that even under imperfect information and bounded rationality, market participants could approximate the equilibrium through iterative interactions, lending support to the robustness of the Walrasian hypothesis under real-world conditions, whilst reaffirming the power of the "invisible hand" within a CDA. This in turn opened up avenues for research into the CDA using participants with limited knowledge.

2.1.2 Modelling the exchange

There are many researchers and corporations with a vested interest in modelling the financial exchange accurately. For the former, it will lead to new avenues of research into the market mechanism with an additional layer of performance guarantee. For the latter, it will translate into higher profits, through a better estimation of market impact, trading cost calculation, and future price movements. Thus, it will be to no one's surprise that most institutional investors and investment banks have their own models to estimate their costs. However, due to the strategic sensitivity of such tools, it is likely that many of the most successful and accurate models are hidden away in the depths of profitable organisations.

A point worth mentioning is that a dynamic model of the exchange is not the only method to measure market impact and costs associated with trading. Analytical methods fitted using empirical data, such as the Barra model (Torre (1997)) and the I-Star model (Kissell et al. (2003)), also provide approximations of trading costs. The former divides market impact into temporary and permanent constituents, both of which depend on the quantity traded and the liquidity conditions in the order book. This model incorporates the "square root law" of market impact, which states that larger volumes face an increasingly higher price resistance as they consume more liquidity, due to the increasing shape of the LOB. This concept is one that we will go into more depth in Chapter 4 of our research. The latter model assumes no permanent market impact but instead decomposes temporary market impact into elements based on the trade's participation rate (the size of the trade relative to the prevailing market volume) and the average daily volume for the asset. Both models are calibrated using historical trade data from various exchanges.

Whilst these models provide versatile and empirically accurate estimates of market impact, they are deterministic in nature once tuned, and function as a lump-sum cost that is incurred on every trade. Our research presents a simulation framework that can deliver realistic market impact from market interactions, as opposed to a deterministic

cost, which in turn allows users to understand and strategise their trades and knock-on impact on the market at a more granular level.

McGroarty et al. (2019) summarises the various methods of modelling the exchange into four categories: theoretical equilibrium models from Financial Economics, statistical order book models from Econophysics, stochastic models from the Mathematical Finance community, and agent-based models from Complexity Science. On top of these categories, we would like to introduce a new method of estimating the LOB using machine learning-based models, to reflect the evolving nature of the field. Findings on various modelling methods, alongside McGroarty's findings, are detailed in the following subsections.

2.1.2.1 Theoretical equilibrium model

The equilibrium approach focuses on liquidity consumption, and is carried out by specifying the participant characteristics and the resilience of the limit order book to reach a market equilibrium. The term "resilience" here refers to the speed the limit order book is able to replenish itself after being hit by a trade (Obizhaeva and Wang (2013)), and is an element of market liquidity. Notably, Kyle (1985) was able to use this method to answer questions such as the speed of private information reaching market price, and the value of the said private information. Likewise, Foucault et al. (2005) categorises the participants with a level of "impatients", and uses the model to show how this characteristic affects market liquidity and resilience in the equilibrium. Goettler et al. (2005) also assigns traders with various strategies and characteristics in a market represented as a stochastic sequential game. However, rather than an analytical solution, they numerically solve for the market equilibrium to generate a time series of trades and quotes. Other approaches Alfonsi et al. (2010); Predoiu et al. (2011) take the dynamics of a limit order book as given, and therefore implicitly use the equilibrium approach to reach conclusions. However, the simplifications which allow these models to equilibrate precisely lead to the limited applicability of these models Farmer and Foley (2009). Rarely can the parameters of these studies be tuned to reflect real-world data, which casts doubts on the practicability of predictions made by these models.

2.1.2.2 Stochastic approach

Stochastic models represent the CDA by modelling the order flows as a random process, usually a Poisson process Abergel and Jedidi (2013); Farmer et al. (2005) as per McGroarty's findings. As Cont et al. (2010) puts it, these models strike a "balance between three desirable features", with these features being the ease of estimation from data, representing empirical properties of the CDA, and analytical tractability. By modelling

each possible action, a participant can make in the CDA as an independent random process, the stochastic model is able to predict the probabilities of various events based on the information from the order book, and effectively capture the dynamics of a CDA. In addition, these models were shown to be promising in capturing the short-term (Cont et al. (2010)) and long-term (Smith et al. (2003)) evolution of the order book. However, the broad description of participants by separating their actions into categories ignores the impact of interactions between participants and is unable to explain the rationale behind trades, and hence provides a sparser explanation of market microstructure. As stated by Johnson et al. (2013), this method of modelling does not "explain the details of the price changes", nor does it "unravel the underlying market microstructure" of market events.

2.1.2.3 Statistical methods

The statistical or the empirical approach looks at the CDA with a data-centric approach. In general, a modelling framework is borrowed from another methodology (e.g. multi-agent simulation using zero-intelligence agents or a stochastic series) and is fitted based on observations from the financial market. Although some research in this field only offers insight into the generic statistical properties of the market, such as the drivers behind large price changes (Doyne Farmer et al. (2004)), others provide a way of representing the LOB using specific parameters that were statistically validated. For example, Bouchaud et al. (2002) models the LOB using the number of "ticks" from the prevailing prices and fits the limit order data using a power law distribution. This distribution is then used to reproduce the CDA in a zero-intelligence (Gode and Sunder (1993)) setting. Other authors use variables such as private valuation (Hollifield et al. (2004)) to achieve similar goals. However, many of these models are too specific to the chosen asset and the environment against which they were tested (Challet and Stinchcombe (2001)), and even those with broader guarantees (Bouchaud et al. (2002)) have little evidence of their flexibility when used in an experimental environment. This is because the emphasis of the statistical approach is on the fit of the model, not the underlying intuition or the explainability of the system or its constituents. We place particular emphasis on the flexibility of the experimental environment, as this is a key component for a dynamic model of the market. All in all, statistical methods' primary focus on replicating the empirical evidence over the causality or the mechanics of the market provides little guarantee on model behaviour when exposed to an artificial exogenous shock, and therefore makes it incompatible with our research goal.

2.1.2.4 Machine learning models

A relatively new strand of modelling the CDA involves Machine-Learning (ML). Like other fields, the predictive power of learning techniques offers new avenues for modelling complex systems such as this. However, many studies that examine the microstructure of the market using this technique focus on predicting the direction of the next market movement (Dixon (2018); Zhang et al. (2019)), rather than replicating the dynamics of the CDA itself. In other words, the focus of most literature is on the profitability of high-frequency trading systems, although there are some exceptions to this (Kercheval and Zhang (2015)), with more research being carried out on replicating the LOB (Ardon et al. (2021); Coletta et al. (2021)).

Although many ML based approaches offer excellent results, a fundamental issue with this technique is the interpretability of the model. In common with the stochastic method, ML-based models offer users a generic overview of the participants' characteristics. However, substantially more data are required to fit the parameters of the model, and they lack even the small amount of feature engineering offered by stochastic models to separate various types of market interactions. Thus, a ML model is unlikely to be able to provide interpretable market dynamics, while relying on the training dataset, making them more susceptible to market and specific and sample-time errors. Consequently, this approach also offers a suboptimal testing ground for our strategies.

2.1.2.5 Agent based models

The Agent-Based Models have contributed a variety of useful ideas to explain the continuous double auction over the past couple of decades. Starting with the influential work of Lux and Marchesi (1999), researchers have used this technique to investigate the dynamic properties of the CDA from a fresh perspective. These models' ability to effectively capture and represent the effect of changing participants and the environment makes this method ideal for simulating interactive algorithms such as Reinforcement Learning, whilst offering flexibility to investigate the LOB on a micro-level to find justification behind each trade. These advantages in terms of explainability and interactivity are the core reasons behind our decision to adopt agent-based models in our research.

2.1.3 Stylised Facts

In order for a model to accurately represent the LOB dynamics, it must behave as expected when left to its own devices, as well as when it is perturbed by an external input. The former behaviour can be identified in the form of stylised facts from the market and can be replicated in the model. There exists a vast amount of literature that looks at the statistical properties of the financial markets to find stylised facts. These properties are

broad claims about the relationship amongst observed market variables (e.g., returns, volume, volatility). Although the exact value of these stylised facts varies across assets, they provide a good starting point for how a modelled exchange should look when left to its own devices. However, most of these stylised facts are collected from the top-level LOB data (i.e. only on trades that were transacted, as opposed to ones that were submitted/cancelled deeper in the orderbook), most likely due to the lack of availability of open LOB data. It is practically impossible to find any amount of freely available Level 2 (or higher) LOB data, so much so that researchers have found a way to generate synthetic Level 3 data (Huang and Polak (2011)), and turned it into a business. Despite this, there are novel sources of data that can be collected to retrieve information on variables such as order arrival rate to paint a better picture of the LOB in our model to increase its realism.

When looking at market-specific stylised facts, there is fortunately an abundant supply of empirical studies across the traditional asset classes. Cont (2001) reported on the properties of the equities market, whilst Ballocchi et al. (1999) focused on the foreign exchange market. Their findings include volatility clustering (otherwise defined as heteroskedasticity of the time series), fat-tailed distribution, autocorrelation of returns, long memory in order flow and returns, and price impacts. Vyetrenko et al. (2021) attempts to replicate many of these findings from traditional asset classes in their work. However, while they successfully replicated a subset of these stylised facts, many properties were not reproduced, especially those regarding asset autocorrelation and the relationship between key variables such as volume and volatility.

2.1.4 Market impact

The stylised facts discussed in the section above examine the market from an observer's perspective. However, an equally important aspect is the market response to trading activity, commonly referred to as market impact. This is an active area of research and is critically important when comparing and benchmarking the performance of individual market participants, as it directly determines the trading costs (often referred to as slippage). Some of the leading work in this domain, such as Donier et al. (2015), provides theoretical insights into modelling market impact but lacks empirical support thus far. On the other hand, models such as the propagator model described by Gatheral (2010) focus on key components of market impact, such as impact and decay, and are supported by empirical evidence. However, these models fail to capture certain aspects of market impact, such as the intentions of participants to trade, which are represented in the concept of a latent order book.

Another way to consider market impact is from the perspective of volume and quote prices versus the market mid-price. When quotes are requested from market makers (this could be in a dark pool or in a quote-driven system), the price offered depends

on the size of the transaction. Larger trades tend to receive worse prices as it becomes increasingly difficult for the market maker to clear their positions without affecting the prevailing equilibrium price. The change in quote price depending on the size of the transaction may also be considered market impact from the perspective of the client.

Most studies on market impact focus on creating models to estimate the costs of a trade rather than simulating the slippage in an interactive environment. One notable attempt to create such a simulated environment is the Santa-Fe model (Farmer et al. (2005)). This work attempts to recreate market impact using a particle-based stochastic model but falls short in explaining gaps between limit orders in the LOB, its inability to replicate market volatility, mean reversion properties and its lag-independent constant market impact result. As Bouchaud explains (Bouchaud, 2018, Chapter 7), there have been attempts to enhance the Santa-Fe model to overcome these shortcomings. However, these enhancements often sacrifice much of the intuitive understanding of the decision-making processes of the participants, reducing the model's explanatory power.

2.1.5 Summary

The empirical studies into CDA have evolved substantially since Smith's experiments on students. The increasing computational power has allowed researchers to adopt numerous computational methods to help understand and reproduce the auction. In addition, parallel advances in data collection and storage capacity have led to a collection of past asset prices that can be used to further progress in this domain. Many studies have since leveraged these advancements and provided meaningful contributions to the field. However, Section 2.1.3 has shown that current empirical studies have been limited to traditional asset classes, and creating a simulation to replicate these stylised facts in a more complete manner is an ongoing area of research. Furthermore, Section 2.1.4 identifies the sparsity of research in the literature that attempts to replicate market impact using market simulations that can interact with strategies.

This points to a gap in the literature within the intersection between effective market simulation of the continuous double auction incorporating stylised facts and market impact, and new asset classes such as cryptocurrencies. Therefore, a study into a market simulation that is designed specifically for a cryptocurrency, such as Bitcoin, that can replicate stylised facts and market impact observed in the market will contribute meaningfully to the market simulation community. Although there are many means to achieve this goal, as discussed in Section 2.1.2, the model intuition and interpretability can be emphasised and enhanced substantially by adopting the agent-based technique. Consequently, the next section of this literature review will focus on the benefits of the agent-based model, as well as previous work done by the agent-based community in the area of market simulation.

2.2 Agent Based Modelling (ABM)

An Agent-Based Model (ABM) is a method for simulating the actions and interactions of autonomous agents on a microscopic level. By tweaking individual or environmental parameters, it allows the user to gain an explanatory insight into the behaviour of each agent, as well as the overall dynamics of the system. This makes it an ideal platform for testing and validating hypotheses in a dynamic environment involving various participants, such as a CDA. In the context of a CDA, not only does the model offer detailed information on each participant, but the environment allows testing for the counterfactual - an incredibly powerful tool that is unavailable in the real world. This approach extends the field of small-sample experimental economics, pioneered by the likes of Vernon Smith (1962), to a generalised and reproducible setting. Although the field has seen some research on small-scale simulations in the last century Gode and Sunder (1993), the rapid increase in the availability of processing power has led to an evolution of small-scale computation into large parallel simulations which can replicate the real world with much greater accuracy.

Leveraging on ever-increasing computational power, financial research using agent-based techniques has been on the rise since the turn of the millennium. Darley et al. (2000) used ABM to look at the market volatility that followed a change in the minimum tick of an exchange, whilst looking at the response of rudimentary reinforcement learning market makers. Wang and Wellman (2017) used ABM to explore the effects of fake orders to "spoof" the market, much like the flash crisis of 2010 (Aldrich et al. (2017)), and its impact on prices and other market participants. Another study by Wah et al. (2017) leveraged the environment's ability to generate counterfactual results to show the impact of a market maker on the CDA's allocative efficiency. A further study worth mentioning is the work by Das (2008), where the speed of a market's price discovery is scrutinised under various types of market makers.

More recent studies have used ABMs to provide experimental evidence for theoretically sound market design improvements that could benefit participants in a financial exchange. Budish et al. (2015) showed that the CDA by design incentivises the use of expensive speed-enhancing technologies, which not only lead to reduced liquidity but also impose social costs on market participants in the form of wider spreads and higher slippage. These costs arise from the high-frequency trading arms race, where traders compete to gain a speed advantage over others to "snipe" stale quotes before prices can fully adjust to new information. To address this inefficiency, Budish *et al.* proposed the Frequent Batch Auction (FBA), a discrete-time trading mechanism that fundamentally alters the incentive structure of the exchange by discretising order matching over short time intervals. Their analysis showed that FBA eliminates mechanical arbitrage rents, significantly reducing the incentive for traders to invest in expensive ultra-low latency infrastructure. In a follow-up empirical study, Aquilina et al. (2020) used proprietary

message data from the London Stock Exchange to quantify the real-world impact of such latency arbitrage. They found that races to snipe stale quotes occur roughly once per minute per FTSE 100 stock and account for over 20% of total trading volume, with the top six trading firms capturing more than 80% of race wins. Although individual races involve small monetary stakes, the aggregate cost to market participants is significant—equivalent to a 0.5 basis point "tax" on trading volume and an estimated $5 billion annually across global equity markets. Their findings suggest that eliminating latency arbitrage could reduce the market's cost of liquidity by 17%, reinforcing the case for discrete-time mechanisms such as FBA. Budish et al. (2014) further outlined a detailed blueprint showing how FBA could be implemented in modern financial markets without disrupting existing market functions. Their proposal addressed regulatory constraints, optimal batch intervals, and mechanisms for order matching, demonstrating that FBA is both feasible and scalable in real-world exchange.

Building on this foundation, Aldrich and López Vargas (2020) conducted experimental studies with human-controlled agent-based models to compare market behaviour under CDA and FBA. Their experiments allowed participants to choose whether to invest in speed-enhancing technology in these two auction formats, with the choice of participating as a market maker or as a "sniper". The results provided strong empirical support for FBA, showing that it improves social welfare by reducing unnecessary expenditures on low-latency infrastructure and mitigating adverse selection costs for liquidity providers, whilst reducing the spread in the market to lower slippage for market participants.

Further exploring these findings with agent-based models, Savidge and Cliff (2023) conducted simulation studies using their own Bristol Frequent Batch Stock Exchange platform to analyse how various algorithmic trading strategies, previously studied in CDA environments, adapt to FBA. Their research used established trading algorithms from CDA experiments (ZIC, ZIP, GDX, GVWY, AA and SHVR) to compete in a FBA. They found that the dynamics changed in the FBA, and the performance of SHVR agent that represents the High-Frequency Trader (HFT)s did indeed fall. However, a small adjustment to the SHVR algorithm to incorporate the nature of FBA led to the algorithm being profitable and outperformed other intelligent agents (AA and GDX), raising questions about the effectiveness of FBA for reducing HFTs and inviting more investigation into the matter.

In the domain of using ABM to explore market impact, Church and Cliff developed an agent-based simulator called BSELD (**B**ristol **S**tock **E**xchange with **L**it and **D**ark venues, Church and Cliff (2019)) that extended the existing Bristol Stock Exchange (Cliff (2018)) to replicate the hybrid market structure of the London Stock Exchange's Turquoise platform. The Turquoise venue operates a dual-market, comprising a fully transparent lit order book (Turquoise Lit™) and a midpoint-crossing dark pool (Turquoise Plato™), where block orders can be matched anonymously at the prevailing midprice.

To model this, the authors included both the lit and dark trading venues in the simulator, with routing logic to either of the exchanges based on order size. They show the importance of incorporating the dark pool by developing a novel agent type, the ISHV trader, which adapts its quoting behaviour based on orderbook imbalance weighted mid-price (the authors call this "microprice"). Using this, they show that in a traditional lit pool, these ISHV agents react to large limit orders (block orders) and demonstrate the impact limit orders can have on the market prior to any execution occurring, and argue the case for the hybrid lit-dark market structure. In addition, they also discuss the plumbing of the reputation-based access to the LSE's Turquoise platform London Stock Exchange (2024) and show their simulation of agents' "reputation score" evolution that enable enhanced price discovery through the "Block Indication" system. This work in market impact and the ISHV agent was continued further by Cliff (2024) when he introduced the PRZI (**P**arameterised **R**esponse **Z**ero **I**ntelligence) trader. PRZI looks to generalise and mimic the family of Zero-intelligence traders using a single strategy parameter $s \in [-1, +1]$ that governs how aggressively or passively an agent quotes prices subject to its profitability and "highest plausible price" (and lowest) constraint calculated based on price observations. This parameterisation allows PRZI agents to approximate the behaviour of various strategies such as SHVR, GVWY, or ZIC. One of the discussed variants of PRZI is an adaptive strategy, IPRZI, where the strategy parameter s is dynamically adjusted based on information from the live orderbook imbalance, using the same microprice function as before. This allows IPRZI agents to adjust prices in response to changing liquidity conditions on either side of the book, leading to market impact being observed in the market from limit orders that are placed on the top level.

The increasing interest in the application of ABMs in finance is not unique to academia. Global financial institutions, notably JP Morgan, have recently made substantial contributions to the field as well. They look at a variety of topics, starting with opponent modelling (Mahfouz et al. (2019)) that shows that agents equipped with opponent modelling capabilities achieve higher success rates in auctions. They use maximum likelihood estimation to infer the distribution of participating agents and their pricing policies, to then optimise their own pricing strategy. When comparing this agent's performance against agents with fixed models or random actions, the authors showed material outperformance of their opponent modelling agent. Another interesting contribution was their research on market making (Ganesh et al. (2019)) in a dealer market. They simulated an over-the-counter market with multiple market makers acting as dealers, and used a reinforcement learning method (Proximal policy optimisation) to learn to show optimal bid/ask spreads and the fraction of the book to hedge. Again, they showed that this agent outperformed both a model-based agent as well as a random agent in their simulation.

Similarly to market simulators, it is likely that most of the institutional market participants have something to offer to the ABM community. However, the nature of the

industry is not one that fuels cooperation, and thus it is likely that many interesting ideas and techniques will remain confidential.

2.2.1 Agent Categorisation

To apply agent-based modelling (ABM) to a financial exchange, participants must first be categorised into distinct agent types. From a real-world perspective, participants in an exchange can be broadly divided into issuers, investors, and market makers.

Issuers are entities whose financial assets are publicly traded. This category encompasses a range of organizations, including major corporations such as Apple and Amazon in the US, as well as BP and National Grid in the UK. It also includes national governments and public institutions, such as universities, when they issue debt in the market. Issuers interact with the market through both direct and indirect listings, exerting significant influence on market dynamics. Their primary objectives often include raising capital, consolidating ownership, or satisfying investor expectations.

On the other hand, investors can take many forms. Institutional investors include pension funds, hedge funds, high-frequency trading firms, and others. Private investors are even more diverse, ranging from retired traders to overconfident graduate students. Although profit generation is often the primary objective for investors, it is far from the only consideration. Many investors operate with a broader set of criteria, such as risk-reward preferences (using measures such as the Sharpe ratio and the information ratio), value-at-risk (VaR) or maximum drawdown thresholds, or specific investment goals such as capital growth or income generation. Some investors may prioritise capital growth, focusing on increasing the value of their investments over time, while others may seek income-based investments that generate steady returns, such as dividends or interest. Environmental, Social, and Governance (ESG) considerations are also becoming increasingly prominent in shaping investment decisions. The impact of investors on the market varies significantly depending on their size, strategy, and priorities. For example, an institutional investor such as BlackRock generally has a substantially larger influence on the market than an average day trader working from their bedroom.

The final type of participants are the market-makers. These intermediaries simultaneously offer to buy and sell a given security, profiting from the margin between their buy and sell prices whilst providing liquidity to the market. Their role is crucial for maintaining a smooth flow of orders within a CDA, thereby enhancing the efficiency of capital markets.

To model these heterogeneous participants within an agent-based framework, they are categorised into three distinct groups: market-makers, fundamental agents, and technical agents. This approach is similar to the "fundamentalist" and "noise trader" dichotomy used by Lux and Marchesi (1999), but with the addition of a third category

of market-makers representing liquidity providers. These three types of agents, which are essential for constructing a robust agent-based simulation, will be explored in detail below.

2.2.2 Market Making agents

Market-making agents offer to simultaneously buy and sell the underlying asset. The gap between their buy and sell prices is called the "spread", and if a transaction occurs on both sides, this spread will correspond to the agent's profit margin per unit. However, suppose the market-maker is unable to transact the same units of the buy and sell orders. In that case, they are exposed to market fluctuations and remain at risk until they can find a counterparty to clear their inventory.

As a profit-maximising agent, the market maker has to manage conflicting priorities. In order to minimise their exposure to the market, the market maker looks to transact as frequently as possible by offering an attractive price on both their buy and sell orders. This causes a downward pressure on the size of the spread. At the same time, in order to protect against market movements eroding away the profit margin and potentially leading to loss-making trades, the MM looks to maintain a buffer between the buy and sell price, leading to upward pressure on the spread. As a profit maximising agent, the MM also keeps track of the price elasticity of the asset to assist with their pricing decisions. Depending on the elasticity, there can be either an upward or a downward pressure on the spread. A successful MM is able to negotiate between these conflicting constraints to generate profit. Some previous studies on managing and optimising these constraints are detailed in the following subsection.

2.2.2.1 Optimal control based traders

The optimal control approach is one of the earliest methods applied to automated trading, including automated market making. Notable foundational works in this field include Garman (1976), Ho and Stoll (1981), and Amihud and Mendelson (1980), all of which study a monopolistic market maker (an environment where all transactions go through a single market-maker who makes the market) operating in a stochastic demand environment.

Garman (1976) introduces one of the first formal models of market microstructure by treating order arrivals as Poisson processes and analysing the behaviour of a central dealer. His framework characterises what he calls the "temporal microstructure" of markets and derives conditions under which the market-maker can avoid inventory-related losses, linking inventory dynamics to quoting strategies.

Ho and Stoll (1981) extended Garman's work by formulating the dealer's problem as a stochastic optimal control problem, where the market maker maximises expected utility in the presence of uncertainty over both execution probability (i.e., whether a counterparty transacts against the dealer) and asset return volatility. Their model produces theoretically optimal bid and ask quotes that dynamically adjust based on the market-maker's inventory, risk aversion, and the history of asset returns. This results in wider and more skewed bid/ask quotes when volatility is high or when the dealer holds a large inventory.

Lastly, Amihud and Mendelson (1980) focused on deriving closed-form expressions for optimal prices in environments where a monopolistic dealer adjusts quotes in response to inventory levels. They show that bid and ask prices move asymmetrically with inventory changes, and that this behaviour induces serial correlation in the market price. Their work highlights quote skewing as a critical tool to manage inventory, and not simply a method deployed for price discovery.

More recently, Avellaneda and Stoikov (2008) furthered the work by Ho and Stoll, and used dynamic programming to solve the assumed utility function of a market maker. This assumed function focused on inventory risk, which was used to determine the optimal pricing strategy. The authors ran a simulation using this utility function in a stochastic LOB environment with varying risk profiles to show the mean-variance trade-off between the discussed "inventory" strategy and a benchmark "symmetric" strategy. This work was then furthered by Guéant et al. (2013) who again focuses predominantly on the inventory risk of the market-maker, but transforms the partial differential equations used to describe the market-maker's utility into a system of ordinary differential equations (ODEs) to simplify the computation. Avellaneda and Stoikov (2008) remains a benchmark in many industrial settings to this date.

2.2.2.2 Reinforcement learning based traders

The first attempt to apply reinforcement learning in a market-making context was Chan and Shelton (2001). They used Monte-Carlo, actor-critic, and State Action Reward State Action (SARSA) algorithms using an epsilon-greedy policy (see Section 2.3 for details of reinforcement learning algorithms) to train their market-making agent. Their environment included a basic simulated market with informed and uninformed traders, as well as a monopolistic market maker quoting prices. The state space included variables such as the agent's inventory, order imbalance, and market quality, and the action space was kept even smaller to discrete changes to the bid and ask quotes. Despite operating under partial observability and with limited state-action complexity to avoid the need for function approximators, this market maker successfully converged to near-optimal quoting strategies that responded appropriately to order flow. Their results showed that RL algorithms could learn policies that balance profitability and inventory control,

although the performance of their algorithm was limited due to the simplicity and the low-dimensionality of their state-action space.

Since then, several studies (Della Penna and Reid (2011); Guéant and Manziuk (2019); Lim and Gorse (2018); Spooner et al. (2018); Zhao and Linetsky (2021); Wang et al. (2024)) have explored the market-making problem using RL techniques, driven by the growing popularity of machine learning. Although the choice of reinforcement learning algorithm and the method of approximating the state space vary across studies, the market-making problem is generally framed with the limit order book as the state space and order submissions as the action space. As a result, once the learning algorithm and the approximation method are selected, the reward function becomes the main tool to tune the algorithm to achieve the desired trading outcome.

To give details, Spooner et al. (2018) designed their agent using TD Learning and tiling approximators with a dampened reward function on profits. Tiling approximators are a type of function approximation that discretises the continuous state space into overlapping tiles, allowing the agent to generalise across similar states. This is particularly useful in high-dimensional state-spaces like the limit order book. The dampened reward function was necessary to prevent the agent from favouring a linear profit-based reward, which encouraged it to hold profitable inventory positions instead of making markets efficiently.

In contrast, Lim and Gorse (2018) used Q-Learning with no state space approximation to train their market-making agent using a reward function based on Constant Absolute Risk Aversion (CARA). CARA models assume that an agent's utility depends exponentially on wealth, which reflects a constant level of risk aversion irrespective of wealth levels. This approach helps discourage excessive risk-taking by penalising high inventory levels or exposure to adverse price movements, aligning the agent's behaviour with that of a risk-averse market maker. Both these papers highlight the importance of shaping the reward function effectively to encourage desired behaviour and, in turn, to develop effective market-makers.

Zhao and Linetsky (2021) introduce the "Book Exhaustion Rate" (BER), a metric that quantifies the rate at which liquidity at the top of the order book is consumed by incoming market orders. They use BER as a proxy for adverse selection risk (the likelihood that prices will move unfavorably for a market maker after their limit order is executed). By incorporating BER into the state space of a reinforcement learning-based market-making algorithm (in their case, a policy gradient approach), they demonstrate that this feature engineering enhances the agent's ability to mitigate adverse selection, resulting in improved market-making performance.

Another contribution was made by Gašperov and Kostanjčar (2021). Here, the authors introduced a deep reinforcement learning framework for market making, incorporating

several innovations. First, instead of traditional gradient descent, they trained their neural networks using a genetic algorithm, improving stability and exploration. Next, they engineered the state space by feeding predictive signals for the price range and trend strength, while refining the action space to allow continuous control over both the agent's spread and skew. Lastly, they introduced an adversarial agent that directly perturbs the market maker's quoted prices, enhancing the RL model's robustness against market disruptions.

Guo et al. (2023) tackled the market-making problem by opting to eliminate the need for manual feature engineering. They proposed a deep RL framework in a HFT setting that uses a convolutional neural network with attention mechanisms to extract features directly from LOB snapshots. The extracted features are combined with handcrafted "dynamic market state" variables to form the agent's state space observations. In addition, they design the market-making task using a continuous action space, allowing the agent to quote arbitrary prices rather than selecting from fixed price levels. When tested on data from the Shenzhen Stock Exchange, their agent outperforms baseline models (including random, fixed quoting, classical control, and other feature-engineered RL agents) in spread capture, inventory control, and Sharpe ratio.

Zheng and Ding (2024) looked at the application of reinforcement learning, again in high-frequency market making, focusing on the frequency of observing and interacting with the limit order book. They present intuitive findings which show higher observation frequency leading to a more accurate estimation of market dynamics, albeit at increased computational cost. The authors formalise this by showing that as the time increment between observations is reduced and falls below a threshold, the optimal strategy from the continuous model can be recovered using a discrete model.

Lastly, a recent work by Wang et al. (2024) applies adversarial reinforcement learning to improve the robustness of the market-making strategy. They design three types of market makers: one that always quotes both bid and ask, another that can choose between quoting bid/ask and resting, and a third that can choose amongst quoting both sides, quoting only the bid, quoting only the ask, or resting. These agents are trained using both Soft Actor-Critic (SAC) or DQN architectures. These agents interact and learn in a market environment that is driven by a Hawkes process, with parameters such as order intensity and decay rate dynamically adjusted by an adversarial agent to hamper the market maker. The authors show that the 4-action agent trained in a low-volatility regime generalises well to high volatility. However, agents trained in high-volatility environments do not generalise as effectively to low-volatility regimes, indicating that robustness is asymmetric in volatility regime and training in mismatched regimes can lead to subpar performance.

However, all of these studies rely on historical limit order book data or a price series generated from a stochastic process, which cannot accurately capture the liquidity or the

price impact of the testing strategy's transactions. As a consequence of these restrictions, these studies restrict their algorithms to small transactions and assume unrestricted liquidity and zero price impact. Furthermore, while they provide empirical results that compare the performance of their strategies with existing methods in the same data set, it remains unclear which strategy would prevail when tested against others in a live market environment. These limitations constrain the applicability of the proposed market-making algorithms and leave important real-world questions unanswered.

Some of these challenges were partially addressed by a team from JP Morgan, who simulated various market-making strategies in a dynamic multi-agent environment to compare their profitability and realism (Ganesh et al. (2019)). Although their experiment was conducted in a suboptimally populated environment, with no theoretical or empirical guarantees of replicating the real-world dynamics of the limit order book, the study offered a valuable comparison of a reinforcement learning agent and an adaptive agent competing within the same market.

Finally, we acknowledge that this section has focused on the application of RL to market making, without providing context on the algorithms themselves. A comprehensive overview of RL algorithms and their learning paradigms will be provided in Section 2.3.

2.2.2.3 Online learning based traders

Another interesting technique used for market-making is online learning. In 2013, Abernethy and Kale developed a market-making agent that utilises online learning to make markets (Abernethy and Kale (2013)). By using a regret-minimising algorithm that dynamically adjusts its bid-ask spread to maximise profit, their agents were able to operate effectively on historical stock market data. Some of their algorithms, such as Market Making using Multiplicative Weights (MMMW), achieved impressive results in simulation. However, many of the assumptions underpinning their trading environment, such as zero market impact and unlimited transactions between price movements, were highly unrealistic and failed to account for the dynamics of the CDA.

2.2.3 Fundamental agents

The market participants we refer to as fundamental agents, comprising both issuers and investors, derive their private valuations for a security from external information such as business insights, fundamental beliefs, and projections regarding the entity's performance. These agents choose to enter the market to buy or sell only when the transaction appears to be profitable relative to their private valuation. By simplifying these agents' behaviour down to an exogenous price input and a profitability constraint,

they can be represented by one of many "background agents" available in academic literature.

2.2.3.1 Zero Intelligence

The first type of background agent is called Zero Intelligence (Zero-Intelligence (ZI)), proposed by Gode and Sunder (1993). Their research showed that a group of very basic agents, all assigned with individual private valuation, that randomly chooses a bid or an ask price, subject to a profitability constraint, will converge to the market equilibrium in a CDA. This is despite the simplistic nature of the agent, with no capacity to learn or memorise past actions. Although their performance was inferior to that of human participants, the paper showed that due to the market discipline imposed by the CDA, learning, intelligence, and profit motivation were not needed to reach a high level of allocative efficiency.

Cliff (1997) further explored zero intelligence agents and showed that the ZI model only converges to the theoretical market equilibrium when demand and supply schedules are symmetric, for example, when the gradients of the linear supply and demand curves are equal in magnitude but opposite in sign. He suggested that agents require a bit more than zero intelligence to reliably reach the equilibrium. To address this, the Zero-Intelligence Plus (ZIP) algorithm was proposed, giving agents a simple form of memory and the capacity to adjust their price quotes according to a profit margin that is constantly being updated. Although we provide an in-depth explanation of the ZIP algorithm in Section 3.1.2, an overview of the ZIP update process is explained below.

The conditions for profit margin update depend on recent trade outcomes (e.g. whether the previous order led to a transaction) and on the difference between the agent's current quote and its underlying valuation (Cliff, 1997, Section 6.1). ZIP uses eight parameters to tune how each agent's profit margin evolves over time, guided by the Widrow–Hoff "delta rule". This rule incrementally adjusts an agent's profit margin to reduce the difference, or the error, between the agent's current quote and their target price inferred from recent market activity. Each margin update is proportional to this error, scaled by a learning rate, and includes a momentum ("stickiness") term that stabilises learning. Of the eight parameters, the first six define upper and lower bounds for three variables: the agent's initial profit margin, its learning rate, and its momentum coefficient. Each agent draws its specific values for these variables uniformly from their respective ranges, creating heterogeneity in the ZIP agent population. The final two parameters specify the bounds for a small random perturbation applied to the target price during each application of the Widrow−Hoff rule, reflecting the uncertainty in how each agent formulates its target price. This mechanism allows ZIP traders to iteratively refine their profit margins and improve overall market convergence.

In his subsequent papers, Cliff optimises these eight parameters using a genetic algorithm (Cliff (1998)), then extends the eight parameters to 60 (Cliff (2009)) by first changing the small random perturbation parameter to be unique to each agent, and letting them sample from a range, similar to the other variables. This initially extends the parameters to 10. Then each of the six individual cases in the original ZIP algorithm (Cliff, 1997, Section 6.1) is tuned independently, multiplying the number of parameters by 60. The new ZIP60 algorithm when tuned via a genetic algorithm outperformed the original ZIP algorithm in terms of equilibriation and remains one of the most competitive background agents to date.

Although ZI traders operate with minimal intelligence by submitting orders randomly, their built-in profitability constraint ensures that they only place orders consistent with a private valuation, mirroring the decision-making process of fundamental agents. While ZI traders do not capture the full nuance of human reasoning involved in the fundamental investment process, they serve as an effective abstraction for modelling the core behaviour of market participants who base their actions on exogenous valuations.

2.2.3.2 Assignment-Adaptive

An agent worth mentioning is the Assignment-Adaptive (ASAD) agent by Stotter et al. (2013), which attempted to enhance the ZIP agent further, especially in market shock scenarios. This agent keeps track of a "shock indicator", calculated using ordinary least squares regression over a time series of the last 20 prices. Depending on the size of the shock, the agent's profit margins are adjusted using this indicator to yield a different profit margin to the original ZIP algorithm. The experimental results showing the agent's reaction to a market shock seemed promising, and the ASAD agents were able to outperform the ZIP agents in a homogeneous market with very little overall profit lost. However, it was unable to outperform ZIP agents in a heterogeneous market containing both ZIP and ASAD, as ZIP agents were able to quickly adjust to the new price signal provided by the ASAD algorithm.

2.2.3.3 GD (Gjerstad and Dickhaut)

Another background agent is called the Gjerstad and Dickhaut (GD) algorithm, named after its inventors Gjerstad and Dickhaut (1998). This strategy maps each bid or ask price with a probability of being accepted and keeps track of these probabilities. Each time a new trade occurs or a new bid or ask is posted, the agent records whether a bid or ask was accepted, and updates its probabilities (otherwise known as the belief function) based on the number of orders accepted and remaining at or better than each price level in the belief function, as well as orders that are rejected at or below each price level. This belief function is smoothed such that the probabilities of being accepted are

monotonically increasing toward the more competitive price at each update. Using this belief, the agent then submits a limit order that maximises their expected profits. This algorithm was modified slightly by Tesauro and Das (2001) and named MGD (Modified Gjerstad-Dickhaut) to trade on a real-time basis. It retains the idea of a belief function from the original GD algorithm, but enables multi-unit trading and incremental updates to belief functions on the go. A few years later, Gjerstad (2007) introduced the Heuristic Belief Learning (HBL) algorithm that also improved the GD algorithm by modifying the timing of the algorithm's order placement and the decision rule of placing the limit order to be less aggressive and closer to the previous transaction price.

However, the variations of the GD strategy only looked to optimise immediate profits, without considering the algorithm's cumulative performance over a longer time horizon. Tesauro and Bredin (2002) addressed this issue in their Gjerstad Dickhaut eXtended (GDX) algorithm (before the advent of HBL), which utilises dynamic programming to optimise intertemporal decision-making. This paper builds on the GD agent by including a time component to the state space. This is accomplished by calculating the expected number of remaining trading opportunities, which can then be used for discounting purposes. A value table, $V(x, n)$, is used for this purpose, and is updated using backward induction, starting from the remaining trading opportunity of 0. From here, the set of feasible values of the agent's holdings is iterated through, and the value table is updated using the sum of values attributed to trading and not trading at the given time, as well as the returns on the agent's holdings. The backward induction is necessary for earlier decisions to incorporate their discounted future values. Once the value function is fully updated at the beginning of each agent's decision-making process, it is used to determine the price and quantity of the upcoming action. Tesauro and Bredin also compared the performance of GDX against both GD and ZIP algorithms with the success criterion of a larger average surplus. As shown by the authors (Tesauro and Bredin, 2002, Figure 3, 4), the GDX outperformed GD as the discount parameter was increased (i.e. as the value of future actions increased) as well as the ZIP algorithm.

2.2.3.4 Adaptive-Aggressiveness

The Adaptive-Aggressiveness (AA) agent by Vytelingum et al. (2008) is designed to learn from the market history to adjust and adapt its strategy to the latest market conditions. As the name suggests, the strategy varies its level of aggression to control its level of activity in the market. An aggressive trader looks to improve the probability of transactions occurring by submitting an offer close to or better than its internal estimate of the competitive equilibrium. On the other hand, a passive trader does the opposite and is more inclined to make fewer profitable trades with a large profit margin between their offer and the estimated market equilibrium. The algorithm has two actors within it. The trading component, which combines the current estimate of the market equilibrium and

the agent's aggressiveness level to submit a bid or an ask, and the learning component, which updates the aforementioned parameters used by its trading counterpart. The equilibrium is estimated using a moving average with an element of discount, whereas the aggressiveness is adjusted using three parameters. First is a distinction between intra- and extra-marginal agents. A participant is intra-marginal if their limit price is better than that of the competitive equilibrium, whereas an extra-marginal agent has a limit price worse than the competitive equilibrium. Next, there are two short and long-term learning parameters. The short-term learning is done by immediate adjustment after a market order. This adjusts the agent quickly to the prevailing market condition using the Widrow-Hoff algorithm. The long-term learning instead looks at the market volatility over time and adjusts the agent's aggressiveness accordingly.

Note that in a perfectly efficient market, extra-marginal agents should not be able to trade, as their inferior limit prices will reduce the total gains across all participants. This algorithm showed higher levels of efficiency than both the ZIP algorithm and the GDX algorithm and outperformed both when tested against market data (De Luca and Cliff (2011b); Vytelingum et al. (2008)). However, recent research by Snashall and Cliff (2019) revealed that upon extensive testing in a more realistic environment, the AA agent significantly under-performed against other much simpler agents, including ZIP and a "shaver" agent, which is designed only to undercut existing orders if profitable. This opened up the question regarding the "best" background agent once again.

2.2.4 Technical agents

Technical agents represent only investors. They do not have an underlying fundamental belief in the asset price. Instead, they reach a private valuation by examining the details of the market microstructure of the product. This can be anything from the price, volume, volatility and others. Like the fundamental agent, when the agent's private valuation is at odds with the prevailing market price, they choose to participate in order to maximise their profits. Due to these differences, the technical agents tend to trade more frequently on a shorter horizon than the fundamental agents.

2.2.4.1 Momentum

The momentum trader makes their trading decisions based on a belief that asset prices have an element of inertia. In other words, these agents buy assets on an upward trajectory expecting them to continue rising, and sell assets with the opposite conditions. This idea was first explored by Jegadeesh and Titman (1993), who looked at the strategy on a 3–12 month timescale. Their findings were solidified by Rouwenhorst (1998), which showed that an internationally diversified portfolio using the momentum strategy returned around 1% a month, irrespective of the market.

2.2.4.2 Mean Reversion

The mean reversion strategy, often referred to as the contrarian strategy (Keim and Madhavan (1995)), was introduced by De Bondt and Thaler (1985, 1987). This paper provides a contradiction to the momentum strategy, suggesting that past losers significantly outperform past winners. It builds on the idea that market participants overreact to recent data, leading to excessive optimism/pessimism, leading to an over/under valuation of the underlying asset, which this strategy exploits for profit. Their empirical study using data over five and a half decades of asset returns shows that the 50 most extreme losers substantially outperform the 50 most extreme winners over a five year period. In addition to academic evidence looking at long term viability of this strategy, empirical evidence from the industry suggests that the strategy is viable on a shorter timeframe, taking advantage of seasonality such as overreaction in January, followed by price corrections over the following months.

2.2.4.3 Noise traders

There are many other types of technical indicators that some traders consider prior to making a trade. A good summary of these technical strategies and agents is provided by Larsen (2007). However, it is impractical to add all these agents individually into a multi-agent model. Therefore, the activities of these other agents, alongside those who act without rationale or information advantage, will be viewed collectively as "noise" for the purpose of our research. This concept of noise traders was explored in detail by De Long et al. (1990), and has consistently appeared in market models to the present day (McGroarty et al. (2019)). In De Long's paper, the noise trader prices themselves using a combination of price fluctuation, deviation from fundamental asset value, and the risk premia associated with noise, following the idea that irrational traders' beliefs can move prices away from fundamentals in a persistent way, and that rational traders must demand a risk premium to bear the uncertainty introduced by these mispricings. On the other hand, McGroarty borrows a zero-intelligence-like noise agent from Cui and Brabazon (2012) (that uses ZI agents with heterogeneous trading frequencies and order placement strategies) to participate in the auction.

2.2.5 Environment

In order to experiment with the various types of agents, a robust testing environment is necessary. Alongside the development of trading agents, researchers have also developed and enhanced trading platforms to replicate a continuous double auction. These environments allow researchers to test strategic interactions, study emergent behaviour, and evaluate both macro and micro level outcomes. With the ever-increasing supply of

computational power, these environments have improved in terms of flexibility and complexity over time. Some of the better-known environments used by researchers in the field will be described in this section.

2.2.5.1 Project PLATO

One of the earliest environments to simulate the CDA was the PLATO (Programmed Logic for Automatic Teaching Operations) system, developed in the 1970s. Vernon Smith leveraged PLATO's network terminals to conduct real-time market experiments with human "players". These experiments formed the basis of experimental economics and demonstrated that decentralised trading agents could achieve rapid convergence to competitive equilibrium prices even under limited information.

In his 1980 paper, Smith (1980) presented one of the first experimental analyses of decentralised market mechanisms implemented on PLATO. The study showed that market participants, when given the opportunity to interact repeatedly in a CDA setting, could reach allocative efficiency by themselves. This work highlighted the ability of simple market mechanism to yield efficient market outcomes and laid the groundwork for future research on experimental economics and mechanism design. Smith (1982) advanced the foundation of the field by arguing that microeconomic systems could be studied as an experimental science. He formalised core experimental protocols, such as induced value theory, and provided empirical evidence that market equilibria predicted by theory do emerge under controlled experiments. His contribution allowed future researchers to utilise market simulation as a tool to validate economic theory.

These contributions were reviewed in Smith and Williams (1992), which highlighted the findings since Smith's initial work, showing that real-time CDA outperforms alternative mechanisms such as sealed-bid auctions, and emphasised the importance of market design and its role in shaping the efficiency of the market. More recently, Dear (2017) provided historical context on the development of the PLATO platform and highlighted how the platform enabled pioneering experiments in economics, and how the adoption of this technology shaped the rise of experimental economics.

2.2.5.2 Santa Fe Artificial Stock Market

Another early computational study of double auction markets with automated trading agents was conducted by Rust et al. (2018). In their experiment, a set of trading algorithms competed in a controlled environment. Despite the diversity of strategies ranging from very simple rule-based agents to complex optimisation players, the market consistently converged to competitive equilibrium prices with high allocative efficiency. In particular, a relatively simple rule-based strategy outperformed many sophisticated

designs, suggesting that minimal intelligence may suffice for efficient price discovery in a CDA. However, in these experiments, the agents did not adapt or learn over time, and their behaviour was static, as opposed to changing based on endogenous factors.

A more dynamic approach was introduced by Arthur et al. (1996), who developed the Santa Fe Artificial Stock Market. This agent-based model featured heterogeneous, (bounded) rational agents whose forecasting rules evolved through genetic algorithms based on their predictive successes. The simulation was designed with a single risky asset and a risk-free asset, with prices determined via market clearing matching system at each discrete timestep. Their study showed that when the participants' rate of learning was low, the market converged towards theoretical equilibrium. However, when the learning rates were turned up, it led to an increase in complex behaviours including something similar to what the practitioners call "technical analysis", which caused asset bubbles, excess volatility, and trend-following dynamics. This work demonstrated how realistic market phenomena could emerge endogenously from agent interactions, as opposed to external factors.

LeBaron (2002) later provided a technical reconstruction and analysis of the Santa-Fe stock market, detailing its design, learning mechanisms, and empirical properties. He showed that the model could reproduce several stylised facts observed in the real world, including volatility clustering, fat-tailed return distributions, and volume–volatility correlations. LeBaron also highlighted the importance of parameter calibration, especially with regard to the frequency of agent evolution, in determining whether the system converges to a stable regime or enters a persistent state of elevated complexity. His report secured the Santa-Fe model's position as a foundational model in agent-based financial market research and brought to light the power of adaptive agent behaviour in driving complex non-equilibrating dynamics.

2.2.5.3 GEM/MAGENTA

In 2001, IBM researchers combined a distributed system for experimental economic research (GEM) with their agent-based modelling framework (MAGENTA) to create one of the earliest large-scale agent-based CDA simulation platforms (Das et al. (2001)). Although the code is not open to the public, the authors indicated that their platform looks somewhat similar to that of the Trading Agent Competition (Wellman et al. (2003, 2001)). Using this platform, experiments involving 12 agents, belonging to one of ZIP, GD or human, were carried out, with the results showing victory for the algorithms across all six tests. Despite the limited scale of the experiment, this was one of the first works that showed the potential of the agent-based simulation in the context of a CDA.

A decade later, this research was revived and extended by a group at the University of Bristol, who developed and used Open Exchange (OpEx) to study interactions between

human and algorithmic traders. De Luca and Cliff (2011a) replicated the IBM results and conducted human vs GDX experiments, confirming that GDX outperformed both humans and ZIP in a CDA. Subsequent experiments explored into the effects of changing the speed of algorithmic agents on market outcomes using various agent "sleep cycles" (Cartlidge et al. (2012)). This revealed that when AA agents were slowed down to human timescale of 10 seconds per sleep cycle (AA-slow), market efficiency, equilibration, and the performance of human traders improved significantly, when compared to using a much quicker AA-ultra agents operating at 0.1 second frequency. This study was extended further by Cartlidge and Cliff (2013), which showed that the market fragmented as the speed of AA agents increased, leading to these ultra-fast agents disproportionately trading with other algorithmic agents, and human traders with other humans, in what they call a "robot phase transition". Many of these findings were later summarised in a UK Government report, which used these studies as part of a broader effort to understand the implications of high-frequency trading for market fairness, transparency, and stability (De Luca et al. (2011)).

2.2.5.4 Bristol Stock Exchange (BSE)

The Bristol Stock Exchange was developed by Cliff (2018) for the purpose of teaching and research. It provides an excellent open source environment for users to test their trading strategies, without the need to acquire market data. The platform offers an environment which mimics the CDA, and by adding various background agents, they are able to replicate the properties of an LOB. However, due to its original focus towards teaching, as well as its relatively early advent in 2012, it offers limited functionalities in terms of recalling actual market data and agent-specific latencies. Cliff has indicated on the GitHub page for BSE that work on BSE2 is underway to improve on the functionalities of the original environment.

2.2.5.5 ABIDES

ABIDES, or the Agent-Based Interactive Discrete Event Simulation environment, is one of the latest additions to the list of environments used to simulate the financial exchange (Byrd et al. (2020)). This simulator is designed in line with the NASDAQ equity protocols ITCH and OUCH and provides an incredibly robust platform to test and develop trading algorithms and other market models. Not only does the environment provide utilities such as an agent acting as a data oracle allowing it to replay the market on an order-by-order basis, but it also improves on previous platforms by allowing for computational and message latencies to be an impacting factor for participating agents.

2.2.5.6 ExPo

One last environment worth mentioning is the Exchange Portal (ExPo, Cartlidge (2020)), developed at the University of Bristol for a teaching purpose similar to BSE. An advantage of this platform is its ability to allow human participants to take part in the CDA, similar to the original platform used by IBM to pit their agents against humans. However, due to its lack of formal academic representation, most likely due to the development of BSE, this environment will not be considered further for the purpose of this research.

2.2.6 Summary

This section categorised the market participants into three distinct groups (issuers, investors, and market-makers), and divided them into three agent types (fundamental, technical and market-makers). In addition, previous work on each of the agent types, as well as potential testing grounds for these agents, has been summarised. The development of a new simulation platform is outside the scope of this research, as recent contributions in the field now offer a robust and scalable open source environment in the form of ABIDES (Byrd et al. (2020)). Likewise, while new agent types can be developed to represent a wider range of market participants, there is an abundant supply of algorithms that can be used to populate a simulation. Therefore, we focus on developing a methodology to populate and tune existing agents on the ABIDES platform to create a realistic financial exchange.

This novel methodology to tune an agent-based model will enable users to create an interactive environment that can be observed and analysed from a microeconomic perspective. This will further the work done by Vyetrenko et al. (2021) by using a novel tuning method to fit the environment to real-world data. In addition, the new methodology will look to tweak the participating agents to reproduce the expected market impact behaviour, when a trade is made. This, in turn, will allow future researchers to create an interactive asset-specific environment with ease, where various market strategies can be tested.

2.3 Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning which has been in the spotlight for much of the past ten years, though less so in recent days since the limelight was taken over by the progress made in the domain of generative models. This technique builds an agent that can interact with an environment of the modeller's choice. Through

many instances of interaction, the agent "learns" and updates itself to make better decisions through trial and error. This process of learning is carried out using a combination of the currently observed state, the chosen action, and the received reward, usually in a Markov Decision Process (MDP). To navigate through this environment effectively, the agent must first explore the environment, and then start using its knowledge to reap the rewards of its exploration. This is dubbed the "exploration" and "exploitation" process respectively. Getting a good balance between these two will determine how quickly and successfully the agent learns in a given environment.

This technique applies itself very naturally to a limit order book environment, as demonstrated by previous studies discussed above (Abernethy and Kale (2013); Chan and Shelton (2001); Della Penna and Reid (2011); Ganesh et al. (2019); Guéant and Manziuk (2019); Lim and Gorse (2018); Spooner et al. (2018); Wang et al. (2024)). The LOB is a discrete and data-rich environment with clear actions and rewards. To answer this problem, we will also use tools from reinforcement learning to tackle various challenges that come with market making. In order to identify methods best suited for our purpose, we must first look at existing methods and applications within the field of reinforcement learning. We will first discuss the various applications in a human environment and then explore more contemporary techniques and developments. Afterwards, we will delve into domain-specific adjustments to help our model calibrate and look to leverage this domain knowledge to enhance our agent further.

2.3.1 History of Reinforcement Learning

The history of RL can go back to the 1950s, when Samuel developed an algorithm that learns to play checkers (Samuel (1959)). However, the breakthrough with Sutton (1988) which used the term "Temporal Differencing" to learn environments. Temporal Difference learning, or Temporal-Difference (TD) learning, estimates the value of a state by comparing the reward received from the current state to the expected reward from the next state in a sequential manner. This allowed Reinforcement Learning algorithms to solve a wider variety of problems in more complex environments when compared to Monte-Carlo methods, which backpropagates and trains the learning algorithm from the end-of-series rewards.

One of the first applications of TD-Learning was the TD-Gammon algorithm created by Tesauro (1995), designed to play backgammon. By using temporal differencing to train an artificial neural network based on board positions, this algorithm was able to reach a level just below that of the best human players of the time and contributed new unorthodox strategies to the backgammon community. A key improvement this algorithm offers is its ability to learn from the environment without the need for an external party to supply a set of correct moves. This allowed the computer to automatically train its agent against itself to generate more training samples to enhance its parameters.

However, by far the larger contributor to modern-day reinforcement learning was developed by Watkins and Dayan (1992), in the form of Q-Learning. This algorithm, designed to solve sequential tasks, estimates the quality of actions in a given state through the use of "Q-values". Using the TD Learning framework, the algorithm iteratively updates these Q-values based on the difference between predicted and observed outcomes. The update rule incorporates the immediate reward, a discount factor for future rewards, and the maximum Q-value in the next state. This allows the algorithm to learn optimal strategies over time by both exploring and exploiting the specified environment. This algorithm proved to be versatile across various application domains, including games, control systems, and more. However, interest in the field at the time was limited to academics and a handful of practitioners.

Interest in the domain exploded when researchers at DeepMind showed the potential of reinforcement learning by using the Atari games environment in their Nature article (Mnih et al. (2015)). Their use of the now-patented "Deep Q-Learning" outperformed humans on around half of the games available on the Atari platform, without being given any information on basic strategies. Furthermore, the agent's state space was determined using only the visual input which humans receive when playing the games. Following this success, their research expanded into more complex games and reached a milestone with the game Go, where their algorithm defeated various professional players, including the world champion Lee Sedol in 2016. An updated version of this winning algorithm called "AlphaGo" was published in Nature (Silver et al. (2017)), revealing the black box behind their success. This success in defeating top human players in games traditionally considered too complex for machines reached another milestone in 2019 when DeepMind's AlphaStar (Vinyals et al. (2019)) defeated top Starcraft 2 players. Although it did not make headlines like AlphaGo, this achievement was at least as significant, due to the complexity of the game environment. Starcraft 2 is a real-time strategy game with a state-action space that is orders of magnitude larger than even the likes of Go. Furthermore, the environment is only partially observable whilst being played out in real time, forcing the agent to make continuous inferences on the unobservable, whilst being tightly limited in processing time.

More recently, OpenAI carried out some studies into reinforcement learning from a slightly different angle. They looked to create learning agents that can collaborate in a multi-agent environment when tackling complex tasks. A simple example was carried out in an augmented hide-and-seek environment (Baker et al. (2019)), which required agents to collaborate and utilise the environment to the best of their ability to win. This simple, yet very intuitive and visible, study shed light on RL agents' ability to collaborate. Soon after, the OpenAI team added to the long list of machine victories in the human domain, by defeating the reigning world champions at the game Dota 2 using their OpenAI Five algorithm (Berner et al. (2019)). Dota 2 is a game where two teams of five players battle in real time to achieve the objective of defeating the opposing team.

It is similar to a real-time strategy game like Starcraft 2, but with unique abilities and roles assigned to each player, requiring team coordination, collective decision-making, and skilled execution as a group to secure victory.

These breakthroughs jump-started research into reinforcement learning across various applications, including manufacturing, operations research, and finance. Although there exists scepticism surrounding the use of RL in mission-critical operations, due to the trial-and-error nature of the learning process and the "reality gap" between simulation and practical application, progress is being made in fields such as autonomous vehicles. In our work, we reduce this "reality gap" by providing a simulation that captures key features such as market impact, to enable further RL research in financial trading.

2.3.2 Types of learning

There are many types of reinforcement learning algorithms that an agent can use to learn about the environment. However, there are pros and cons to each depending on the use setting. Below, we will discuss the various choices that have to be made, and the trade-offs we face when deciding upon a reinforcement learning algorithm.

2.3.2.1 Temporal Differencing vs Monte-Carlo

The first choice when it comes to modelling an RL agent is between Monte Carlo (MC) learning and Temporal Difference (TD) learning. MC learning tracks all state-action-reward tuples the agent has visited over an episode of the simulation. Then, these tuples are used to incrementally train the agent's parameters, be it the "Q" function or the policy. On the other hand, TD learning is solely based on observed returns at the end of an episode to update the value function.

Consequently, the TD method has a lower variance due to incremental updates, but can potentially introduce biases due to inaccurate estimation of the next state, or through biased bootstrapping of observations. On the other hand, the MC method does not introduce any bias but can lead to very high variance as the length of the episode increases or if the episode includes elements of stochasticity.

2.3.2.2 Model-free vs Model-based

In model-free reinforcement learning, the agent figures out the dynamics of the environment purely through trial and error. By interacting more with the environment, the agent forms a better picture of an optimal interaction strategy, but does not explicitly attempt to learn the dynamics of the environment. On the other hand, in Model-based

Reinforcement Learning, the agent explicitly constructs a model of the environment, including the transition dynamics that describe the evolution of the system from one state to another following an action.

Unsurprisingly, the Model-free approach is easier to use in a complex environment, where the system dynamics is difficult to model, due to features such as dimensionality or stochasticity. However, as the decision-making process is learnt purely through trial-and-error, it is rather sample-inefficient. Conversely, the Model-based approach is able to make more informed decisions by simulating the consequences of its actions using the model of the environment that it has learned, allowing it to be more sample-efficient. However, learning this model of the environment is very challenging, and learning an incorrect model of the world can lead to poor decisions being made by the agent.

2.3.2.3 Types of Model-free Reinforcement learning

As we are working in a simulated environment with a large sample size and complex environment, we are more likely to adopt a model-free approach. There are three dominant branches of model-free reinforcement learning: value function; policy gradient; and actor-critic methods. We will explain these in turn.

Classic value function methods such as Q-Learning (Watkins and Dayan (1992)) and SARSA (Rummery and Niranjan (1994)) are designed to learn the value of a given state, based on the expected rewards the agent can receive from the state, as well as the discounted rewards from future states. This allows the agent to act based on the perceived value of the new reward state, whilst learning the value of the states simultaneously. When this method is given some level of stochasticity when choosing the next action, it can converge to the optimal solution in the environment. However, as the size of the state-action space of the problem grows, the algorithm cannot attribute values to every available state in the environment, as this becomes too costly. Consequently, an approximation function is used to represent the state space. Various approximating functions including linear functions, radial basis functions, and Fourier basis have been used to estimate the value function for the Reinforcement Learning algorithm. However, the most notable progress was made by using neural networks (Mnih et al. (2013)), following the rise in popularity of machine learning. Despite these approximators exhibiting some problems such as a lack of guarantee of convergence, their ability to efficiently explore the space proved incredibly effective, and led to their rapid adoption within the field of reinforcement learning.

Policy gradient methods differ from value function methods by focusing on learning the action probabilities of the state space as opposed to the value of states themselves. Therefore, rather than attributing a value to each state space, the probability attributed to each action in a given state is learned instead. This can lead to an optimal policy that

can deal with stochastic environments, as the algorithm can assign equal probabilities for multiple actions in a given state. An example of a vanilla policy gradient algorithm is REINFORCE (Williams (1992)). This method has a policy function, which has a probability of selecting an action given a state-space input. The agent first goes through an episode of the environment and collects rewards. Once the episode (or a batch) is complete, the rewards are discounted back to each action that the agent has taken, and the policy function is trained to make more of this action if the discounted reward is high or less if the discounted reward is low. At this point, the problem becomes a supervised learning task, where the policy function is trained against discounted rewards to choose actions that yield higher rewards. Unsurprisingly, a popular policy function that gives the probability of making each action uses neural networks to reach its decision and has led to algorithms such as Proximal Policy Optimization (Schulman et al. (2017)).

The final family of model-free reinforcement learning is called Actor-Critic. This method enhances the policy gradient method by reducing the variance of the algorithm's rewards by subtracting a "baseline" reward from the obtained reward after each action. This allows for a smaller, more stable update to the policy function than using the batch methodology explained above in the policy gradient section, due to the incremental updating procedure, as well as a smaller update to the policy, due to the "baseline" variable reducing the magnitude of the update. This leads to a more stable convergence and better algorithmic performance. This "baseline" is calculated using the value function of the state-action space, which itself is trained using a separate function approximator (usually a neural network). This means that the Actor-Critic architecture leverages techniques from both the Policy Gradient method and the Value function method when learning the environment. A good example of an Actor-Critic algorithm is the A3C (Mnih et al. (2016)) developed by DeepMind in 2016. This algorithm uses the value function as the "baseline" and calculates the "Advantage", which is the additional reward received from taking a specific action over taking a random action within the state (the value of this random action is the estimate given by the "baseline"). This value is then used to train the optimal policy for the environment. Here, the policy function deciding on the action taken by the reinforcement learning algorithm is labelled the "Actor" and the adjustments made in the learning process based on the relative reward of each action is called the "Critic".

2.3.2.4 On-policy vs Off-policy

Another factor to consider is the choice between on- and off-policy methods. Like many of the previous trade-offs, this is another trade-off between sample efficiency and learning stability. We will discuss the properties of each policy below.

On-policy reinforcement learning receives experience based only on the actions taken by the agent. One of the most well-known on-policy algorithms is the SARSA model

by Rummery and Niranjan (1994). The nature of the on-policy algorithm means that the only information that the agent remembers and learns from is based on the action it took. In turn, this process requires many sample points for the agent to learn the entire state-action space accurately. In addition, this agent can easily settle in stable local maxima as opposed to continuously seeking a global maximum, especially if the agent's exploration can lead to a large negative outcome. Therefore, when applied in an online setting, an on-policy RL agent is preferred when a mistake can be very costly. Fortunately, the on-policy agent can learn optimal policy over time if the exploration rate is controlled appropriately.

On the other hand, off-policy reinforcement learning (e.g. Q-Learning, Watkins and Dayan (1992)) capitalises on the information regarding the "optimal" action at each point, on top of the action taken. This means that the agent is able to navigate through the state-action space and train itself using the results of the best action, without necessarily making that action. As a consequence, an off-policy agent can learn the optimal policy with a smaller sample size than its on-policy counterpart, as the potential penalty from exploration is lower than that of an on-policy agent. However, the convergence property of an off-policy learner is not guaranteed in many applications, and this type of agent can end up introducing instability as a consequence of updating its parameters only from a sample of the best actions.

2.3.3 Function approximators

If the state-action space is small, the space can be represented in a tabular form, with every possible state-action pair having a value for its own expected future value in the form of a Q-table (value function) or a policy tuple (policy gradient). This is the most accurate method for identifying the environment's dynamics, as every single state will be explored over time, which in turn will result in the agent interacting optimally with the environment. However, this tabular representation is computationally expensive and suffers from the curse of dimensionality. Therefore, it is unrealistic to train a large table representing complex environments such as an LOB, as the processing cost to visit every possible state enough times to train the agent sufficiently is huge. Therefore, approximation functions are used to both reduce the processing requirements and to offer the benefit of generalisation. Approximation functions allow the agent to make informed decisions in previously unexplored territory, based on similar situations it has seen in the past, somewhat like humans. This is a key element behind the success of function approximations in Reinforcement Learning.

There are many ways to approximate the "Q" function or the policy. The most common function approximators include polynomials, radial basis functions, and artificial neural networks. Sutton and Barto (2018) offer an extensive explanation of function approximations in the second section of their book. Below, we will provide background

on some of the most successful Reinforcement Learning algorithms that utilise function approximations.

2.3.3.1 DQN

The most successful value function approximator to date is the Deep Q-Network (DQN), patented by DeepMind. Using artificial neural networks, this approximator can represent any continuous function, making it an ideal tool to be used in an unknown environment. This is especially true in a natural environment, as most phenomena found in nature tend to be continuous. However, the key driving force behind the success of DQN is not the use of convolutional neural networks, but instead its use of experience replay and fixed Q targets.

Experience replay maintains a history of past state-action-reward-state tuples and allows the RL agent to occasionally retrain itself on one of these previous states. By doing this, there is a larger pool of training data available for the agent, making it more data-efficient. At the same time, the correlation between consecutive observations is removed, reducing the variance of the value function update. In addition, by returning to a saved state with current parameters, which are almost guaranteed to be different, it tunes the value function approximator to be more generic and reduces the likelihood of becoming stuck in a local optimum.

The fixed Q target holds the value function as a constant for a chosen number of iterations and only updates it once the iteration counter hits zero. By choosing not to update on every timestep, the RL agent's exposure to noisy input is reduced, making the learning process more stable. Although there is no formal guarantee, this method makes divergence and oscillation much less likely.

The combination of these techniques led to a substantial performance gain and, through it, the Atari 2600 games were mastered using a single set of parameters for all games on the platform.

Soon after the success of DQN, other improvements to the algorithm came in rapid succession. The first enhancement was named double DQN (Van Hasselt et al. (2016)). This adjusted the fixed Q target by keeping two separate instances of the value function, rather than a delayed update to the value function. Although this is likely to lead to a reduction in sample efficiency, it effectively separates the process of choosing the best action and updating the value attributed to the action. This in turn dampens Q-learning's tendency to overestimate the value function and helps the agent find better policies.

Another algorithm that furthered the success of DQN is called prioritised experience replay (Schaul et al. (2015)). This method uses the history of past experiences in a more

effective manner than uniform random sampling that is carried out by the original DQN. By assigning a priority to each tuple in the experience replay that is proportional to the DQN error, past experience is sampled proportionately to the priority. This results in the increased selection of past actions which disagree the most with the current value function, leading to greater sample efficiency and an approximate value function that is better generalised.

The last improvement to the DQN we will discuss is the "dueling" network architecture by Wang et al. (2015). By leveraging on Baird's work on advantage updating (Baird (1993)), the Q function is split into the value of the state and the value of the action by using the advantage function. This augmented structure hastens the learning of the action space, resulting in a faster identification of the most rewarding actions at a given state.

To wrap it all up, Hessel et al. (2018) combined all these enhancements to create an algorithm called "Rainbow". This algorithm showed materially higher performance than any single augmentation to DQN when benchmarked.

2.3.3.2 PPO

Similarly to DQN in the realms of value-function methods, the PPO algorithm (Schulman et al. (2017)) has received a lot of spotlight on the policy gradient side of the equation. Developed by researchers at OpenAI, it uses neural networks to approximate the policy function, as opposed to the state space. However, like DQN, the success comes less from the policy approximation and more from the objective function of the policy update.

The PPO architecture caps the magnitude of each policy update, using a combination of a minimum function and a "clip" function to its policy update. Consequently, the policy can only change incrementally, even when the rewards suggest that a large change should be made. This helps with the algorithm convergence, as these sampled rewards are very noisy and can often overshoot the training process of the policy function. In addition to this, the idea of a "baseline" is also introduced into the algorithm to reduce the size of the rewards (or the "advantage" in this case) that the policy function uses to train itself.

The PPO algorithm is a halfway house to the Actor-Critic family of RL algorithms, with the difference being that there is no explicit value function that is used to critique the policy. However, they are very similar in design and PPO has been shown to outperform state-of-the-art algorithms such as A2C.

2.3.4 Summary

This section of the literature review summarised the history and types of reinforcement learning, the adoption of neural networks as function approximators, and finally the state-of-the-art algorithms that leverage these neural networks to interact with complex environments. Although there are elements that are left out, such as hierarchical reinforcement learning and human-in-the-loop reinforcement learning, as well as technical enhancements made to state-of-the-art algorithms that allow parallel training using multiple nodes, we will conclude the literature review here. This is because we are using RL as a tool to solve the market-making problem, as opposed to pushing the boundaries of the Reinforcement Learning domain. Therefore, we believe that an overview of its history, a brief summary of available methodologies, as well as an explanation of popular methodologies that we may use, are sufficient for the purpose of our research.

One missing element is the summary of progress that has been made within the application domain of market-making, using Reinforcement learning. Gašperov et al. (2021) provides a good overview of progress in this field up to 2021, including the literature that sparked our interest in this domain by Spooner et al. (2018). While we can go through the progress within the domain within this section, we opt instead to leave this part to Chapter 5, where we discuss various approaches to modelling a market-maker using RL alongside our experimental design.

2.4 Bitcoin Market Microstructure

Bitcoin and other cryptocurrencies have emerged as a new asset class with distinct market microstructural characteristics. Unlike traditional financial markets, cryptoassets trade continuously (24/7 without any breaks) across multiple venues, with a diverse mix of participants ranging from teenage retail investors to the most sophisticated high-frequency trading firms. Many of the differences arise due to the lack of regulatory intervention in this market, and as such, the structure of this marketplace presents both familiar and novel dynamics for researchers in this domain.

The study of Bitcoin market microstructure has accelerated since the first cryptocurrency bubble that began in late 2017, with researchers applying and extending tools from traditional asset classes such as equities, FX, and futures market microstructure to examine trading behaviour, price formation, and statistical properties of order flow. The wide ranging literature that has become available since then has been summarised by surveys such as Fang et al. (2022) and Almeida and Gonçalves (2024), which offer useful overview of the research done in this domain. Out of the broad literature available, we focus on two key aspects of the Bitcoin market microstructure: stylised facts and market impact. The former encompasses regular empirical findings in market

behaviour, such as volatility patterns, return distributions, and order book dynamics, whilst the latter looks at how trades influence prices and its implications on liquidity, costs, and market efficiency. Both these elements will assist us in creating an interactive market environment that replicates real-world market behaviour.

2.4.1 Stylised facts

stylised facts refer to empirical regularities observed across assets and time, which have become crucial in understanding market structure and participant behaviour. The literature on stylised facts of Bitcoin and other major cryptocurrencies draws heavily from methodologies used in traditional asset classes, as researchers take increasingly large interest in the domain of cryptocurrencies. The literature on stylised facts we cover in this review looks at similar topics to our discussion in Section 2.1.3 that looked at reproducing stylised facts in simulations, and ranges from fat-tail distribution of returns to U-shape intra-day volume profile observed in the cryptocurrency market.

The evidence for fat-tailed returns distribution that indicates the higher frequency of outsized returns (both positive and negative) leading to fragile prices is provided by Chu et al. (2015), Zhang et al. (2018), and Phillip et al. (2018). They confirm this fat-tailed nature of the crypto market by looking at data across multiple cryptocurrencies. Bariviera (2017), as well as some of the papers above, offer evidence of volatility clustering and the persistence of these volatile periods in the crypto market, and show that while these conclusions are consistent with traditional asset classes, these factors are particularly pronounced in cryptoassets. Zhang *et al.* also provides evidence of decay in autocorrelation and shows that decay is quick in the crypto markets, which suggests shorter memory within the market. This is in line with findings by Bruzgė et al. (2023), who compare cryptocurrencies to tech equities, noting that the former show more pronounced risk characteristics, including faster decays in autocorrelation, higher volatility, and heavier tails.

Eross et al. (2019) offers insight into the intra-day dynamics within the crypto market, presenting a clear U-shaped intraday volume distribution in Bitcoin, similar to that of the equity and FX markets, where activity peaks at the start and end of the trading day. Although crypto trades 24/7, they show that the time of day remains a significant factor in intra-day volumes, with spikes in liquidity and volatility at global market opening hours. We also have contributed to this domain in our paper Cho and Norman (2021), where we collected data on asset returns, autocorrelation, volatility, traded volume, order size, and trading time, as we prepared the environment to fit our simulation to the stylised facts observed in the market. Our results were also broadly in-line with other researchers' findings, and we were the first to present evidence of a positive correlation between returns and volatility in cryptocurrency markets, as well as to document

the power-law distribution of order sizes and the tendency for orders to cluster around round-number quantities.

More recently, with the increasing availability of LOB APIs from leading crypto exchanges such as Binance, studies on the order book have uncovered notable regularities in book shape and behaviour. Angerer et al. (2025) document intra-day variations in depth, imbalance, and spread across exchanges, and how traders can exploit these data to reduce their trading costs. In addition, they find evidence of crypto exchanges influencing the book liquidity of their most liquid pairs by listing more tradable pairs on their platform. On the other hand, Han (2024) used the same order book data to highlight price clustering at round-number levels (e.g., at 00 cents or 50 cents) and draws parallel to the same behavioural bias observed in traditional asset classes. Aleti and Mizrach (2021) also used the order book data to compare it against the regulated futures market on the CME. Although they found stark differences in trade size and liquidity, they found that the crypto market also offered extremely tight bid-ask spread (around 0.03% of the price in liquid pairs). In addition, they found that even a large market order ($1 million) would move the spot by less than 1%, indicating a deep pool of liquidity, and highlighted that sub-millisecond trading activity on these exchanges suggests arbitrage between exchanges by algorithmic traders, leading to shared pool of liquidity across exchanges. Although past studies (such as the work by Urquhart (2016)) found inefficiency in early Bitcoin trading, later works such as López-Martín et al. (2021) report a trend toward greater efficiency in crypto markets, as arbitrage opportunities diminish and liquidity improves following the introduction of algorithmic players. These dynamics suggest that the asset class is maturing and the market structure is increasingly resembling the traditional assets.

2.4.2 Market impact

As discussed in Section 2.1.4, the study of market impact is vital to understanding execution costs, liquidity provision, and the broader implications of price movements within the market. This field has not received much attention within the domain of cryptocurrencies, with some papers skirting around but failing to explicitly model the idea of market impact and slippage during transactions (Genet (2025)), and others offering only broad behavioural perspectives on liquidity or participants following an event (Martins (2024)) or changes to the regulatory framework (Chen et al. (2025)). In general, we were unable to find any studies of market impact that deal directly with the costs and slippages involved with the transaction itself after the 2017 crypto boom.

Prior to the boom, Donier and Bonart (2015) provided one of the earliest and only empirical studies on Bitcoin market impact. Analysing over one million metaorders, they confirm the square-root law of impact in the crypto environment, where price impact

of market orders scale sub-linearly with order size. This result is in line with the findings in equities and futures, suggesting that market impact in crypto follows the same rules of liquidity consumption and orderbook replenishment (for comparisons across asset classes, see Donini et al. (2022)). However, since this study was based on data before the cryptocurrency boom (and sourced from the now bankrupt Japanese crypto exchange Mt. Gox), the market dynamics may have changed since their findings. In the process of creating our market impact simulation (*PRIME*, Cho et al. (2023)), we used data from Binance to verify that Donier's findings still hold after the boom, where the number of participants and the total traded volume have both increased drastically. In addition to providing confirming evidence regarding initial impact, our paper extends the study to show evidence for the reversion of initial impact that follows the propagator model and also shows the behaviour of decay in auto-correlation of order sign.

2.4.3 Summary

This final section of the literature review provides an overview of research that has been carried out in this relatively new field of cryptocurrencies, focused on empirical studies that help us build a better picture of the market as a whole. We divided the review into what we view as "passive" observations in the form of stylised facts that cannot be changed and are beyond the scope of explaining individual orders, and "active" observations in the form of market impact, which relate directly to the participant's trade-by-trade decision making. We found that the former has received a lot of attention over time, however the latter has received little to no love over the same period. Our work contributes to both sides of this research, as a stepping stone to building a realistic market simulation.

Chapter 3

Simulating the Bitcoin exchange

In this chapter, we address the first two research objectives of our thesis (Section 1.2), by presenting a novel methodology for tuning agent-based simulations to replicate stylised facts collected from the financial markets, combining domain expertise with response surface optimisation. We apply this approach to the BTC/USD cryptocurrency market using data collected from Binance. The result is a simulation framework, built on the ABIDES platform, that successfully reproduces key stylised facts (Cho and Norman (2021)). To our knowledge, this represents the first agent-based market simulation explicitly calibrated to reflect the dynamics of a cryptocurrency exchange. In the process, we also provide the first evidence of a positive correlation between returns and volatility in cryptocurrency markets, as well as the first documentation of both a power-law distribution in order sizes and a pronounced clustering of orders around round-number quantities.

We begin our research by initialising the model through the identification and categorisation of market participants into existing agent types available in the literature. We then collect simulated market data from the model and compare it with real-world data obtained from the exchange. Based on this comparison, we tune both the individual parameters of the participating agents and the overall simulation configuration to ensure that the resulting emergent behaviour aligns as closely as possible with observed real-world market observations.

3.1 Market Participants

We start by providing descriptions of participating agents in our simulation in more detail. The agent-based model allows for an unlimited number of heterogeneous agents to be included. Therefore, in an ideal world, we can identify every single market participant and model them into the system to improve the accuracy of the simulation.

However, due to limited information regarding participating individuals and their specific behaviour, as well as our limited computational capacity, there are practical limits on the number of heterogeneous agents. As a solution, we assume that participants fall into one of three broad categories: Fundamental, Technical, and Market-Maker. This is similar to the "Fundamentalist" and "Noise trader" divide that Lux and Marchesi (1999) use, but we explicitly include the liquidity provider, as they serve a distinct purpose in modern financial exchanges. A point to note before describing each agent type is our take on the simulation. As with any complex system, there is always a temptation to add additional features to models to increase their perceived accuracy (or profitability in our particular context). However, this comes at the cost of model complexity, and we believe that it is better to use simpler, more intuitive agents to take full advantage of the ground-up intuition offered by agent-based models. Therefore, whenever possible, we opt for the intuitive, easy-to-explain models, even if they may not be the "best" or the most accurate candidate to represent its category of market participants.

3.1.1 Zero-Intelligence

Many fundamental participants who arrive at a private valuation via exogenous means do not transact in a very sophisticated manner. A good example of these participants is the retail trader, whose execution strategies are relatively basic and incur greater transaction costs than institutional investors. Nonetheless, these traders possess enough intelligence to avoid transacting at prices worse than their private valuation of the asset, thus constraining their transaction prices at or better than their private valuation (less than or equal to private valuation for buyers, and greater than or equal to for sellers). This makes the Zero-Intelligence algorithm (Gode and Sunder (1993)) with budget constraint ideal to represent these participants in our simulation, as this algorithm follows a simple rule that is basic but constrained in a similar fashion (Algorithm 1). Our implementation of the Zero-Intelligence (ZI) algorithm is a slight augmentation of the original algorithm ZI, with multiple units of trade volume placed per order and limit orders placed in the LOB waiting for transaction at the agent's private valuation.

3.1.2 Zero-Intelligence Plus

The more savvy participants execute their trades using execution algorithms that reduce transaction cost versus private valuation, by adjusting prices according to the prevailing market conditions. This is done using either an in-house execution algorithm, through an institutional sales trader, or via other forms of execution specialists. As argued in other papers (le Calvez and Cliff (2018)), we believe that these agents are well represented by the zero intelligence plus (ZIP) algorithm, which is able to adjust their price based on the prevailing market conditions in a similar way to the technical

Algorithm 1 Zero-Intelligence Algorithm

```
Obtain private valuation, V
Query best market ask, A'
Query best market bid, B'
if agent is a buyer then
    if A' exists and V ≥ A' then
        Place a market order to buy
    else
        Place a limit order to buy at price V
    end if
else
    if B' exists and V ≤ B' then
        Place a market order to sell
    else
        Place a limit order to sell at price V
    end if
end if
```

agents. However, unlike the technical agents we describe below in Sections 3.1.3 and 3.1.4, endogenous market data is used only to adjust the profit margin of the agent, rather than contributing to the agent's private valuation. Given this nature of the ZIP algorithm, we use it to represent orders submitted by more sophisticated fundamental agents, such as institutional players. The Zero-Intelligence Plus (ZIP) follows the logic behind the Zero-Intelligence (ZI) model, but incorporates a profit margin when making pricing decisions (Algorithm 2). This profit margin is adjusted dynamically in the market, which removes a degree of naivety from the original zero-intelligence algorithm. Here we follow the nomenclature in Cliff (1997) and define the shout price p_i for each agent i as the price the agent submits to the market after considering their private valuation and profit margin as defined in Equation 3.1.

$$p_i(t) = \lambda_i(1 + \mu_i(t)), \text{where } \mu \in [-1, \infty) \tag{3.1}$$

For an agent with private valuation λ_i, the shout price is controlled by increasing or decreasing the profit margin μ_i. The agent changes the profit margin based on the previous order Q that was submitted to the market, following the set of rules described by Cliff and Bruten (Cliff, 1997, Section 6.1). Here, Q is an array that contains information on the previous order price $q(t)$, whether it was a bid or an offer, and whether it was transacted. The magnitude of the change in the profit margin to obtain the new margin $mu_i(t+1)$ is controlled using the agent's previous shout price $p_i(t)$ and the agent's target price $\Gamma_i(t)$ as shown in Equation 3.2.

$$\mu_i(t+1) = (p_i(t) + \Gamma_i(t))/\lambda_i - 1 \tag{3.2}$$

Algorithm 2 Zero-Intelligence Plus Algorithm

```
Obtain private valuation, V
Query best market ask, A'
Query best market bid, B'
Calculate profit margin, m
if Buyer then
    if V − m ≥ A' then
        Place market order to buy
    else
        Place limit order to buy at price V − m
    end if
else
    if V + m ≤ B' then
        Place market order to sell
    else
        Place limit order to sell at price V + m
    end if
end if
```

The parameters we can control in the ZIP agent appear in the final couple of layers, where the target price of the agent, Γ_i, is updated. It follows a momentum-based update, $\Gamma_i(t+1) = \gamma_i \Gamma_i(t) + (1-\gamma_i)\Delta_i(t)$, with γ_i controlling the stickiness of the target price with respect to the previous target price and a new input, $\Delta_i(t)$, defined in Equation 3.3.

$$\Delta_i(t) = \beta_i(\tau_i(t) - p_i(t)) \tag{3.3}$$

Here, β_i is the learning rate and $\tau_i(t)$ is generated using a function based on $q(t)$, the previous price submitted to the market, $\tau_i(t) = R_i(t)q(t) + A_i(t)$, with $R_i(t) \simeq 1$ and $A_i(t) \simeq 0$ referring to fixed values assigned to each agent i with small random perturbations around 1 and 0 to provide some differentiation amongst ZIP agents.

Before concluding this introduction to the ZIP algorithm, we offer a couple of brief justifications for its use in our simulation. Firstly, unlike its predecessor ZI, the ZIP agent introduces a degree of endogeneity, as described above. This means that the agent's transaction prices are not strictly determined by exogenous factors. As a result, one might argue that such an agent is ill-suited to represent purely fundamental traders. However, the key point is that this endogeneity is used solely to optimise the agent' s margin between the prevailing market price and its fixed private fundamental valuation. In other words, while execution responds to market conditions, the valuation itself remains exogenously assigned. This is analogous to an institutional investor who derives asset valuations through fundamental research but relies on sophisticated execution algorithms to achieve favourable trading outcomes. On this basis, we believe that the ZIP agent is an appropriate choice for our purposes.

Secondly, as discussed in Chapter 2, we acknowledge the existence of other types of fundamental agents that can represent the role of sophisticated institutional traders. A notable candidate is the GDX algorithm (Tesauro and Bredin (2002)), which has been shown to exhibit superior price discovery and faster equilibration compared to the ZIP algorithm. However, these benefits come at the cost of increased computational and memory complexity.

The difference in computational complexity between GDX and ZIP is due to their underlying decision-making frameworks. ZIP agents adjust their bid or ask prices using a simple linear learning rule, based on the most recent market events. Each update requires only a few arithmetic operations, resulting in a computational complexity of $O(1)$ and a total simulation complexity of $O(N)$ for N agents. Because ZIP agents do not attempt to anticipate future market states or optimise intertemporally, their computational requirements remain constant regardless of market conditions.

In contrast, GDX agents employ dynamic programming to optimise expected profit over a set future trading horizon. Each decision involves evaluating multiple inventory states across future time steps and computing the expected pay-off of feasible bid or ask actions at various price levels. This results in a computational complexity of approximately $O(K \cdot T \cdot P)$, where K is the maximum number of units the agent can trade (i.e., the state space of the inventory), T is the discretised planning horizon, and P is the number of price levels evaluated. Therefore, a simulation with N GDX agents yields a total computational complexity of $O(N \cdot K \cdot T \cdot P)$. This is significantly higher than for ZIP, and scales with the granularity of both the price and the state space of the inventory. In highly dynamic markets, where belief functions and price ranges must be updated frequently, the computational burden of GDX agents is further amplified. Consequently, GDX imposes a substantially higher processing load than ZIP, particularly in large-scale simulation.

On the memory side, the requirements of the two algorithms also reflect their design differences. ZIP agents maintain only a small set of numeric values per agent, such as the profit margin, bid/ask price, and the learning rate, yielding a memory complexity of $O(1)$ per agent and $O(N)$ for the full simulation. They do not store historical market data or distributions over prices, making them highly memory efficient. On the other hand, GDX agents maintain an adaptive belief function across a discretised price space and uses a value table to evaluate future rewards across different inventory states. This leads to a memory complexity of $O(M \cdot P)$ per agent, where M is the potential number of units to buy or sell and P is the number of price levels tracked. Although GDX does not explicitly store full trading histories, it maintains this belief function, and as a result, total memory complexity for N agents becomes $O(N \cdot M \cdot P)$.

In Figure 3.1, we illustrate the divergent memory usage profiles of ZIP and GDX agents as the number of agents increases. Beyond this, in long-running simulations where the

price process drifts significantly, the number of price levels P may need to grow in order to capture new market highs or lows. This further increases the memory requirements of GDX agents and hampers their scalability relative to ZIP.

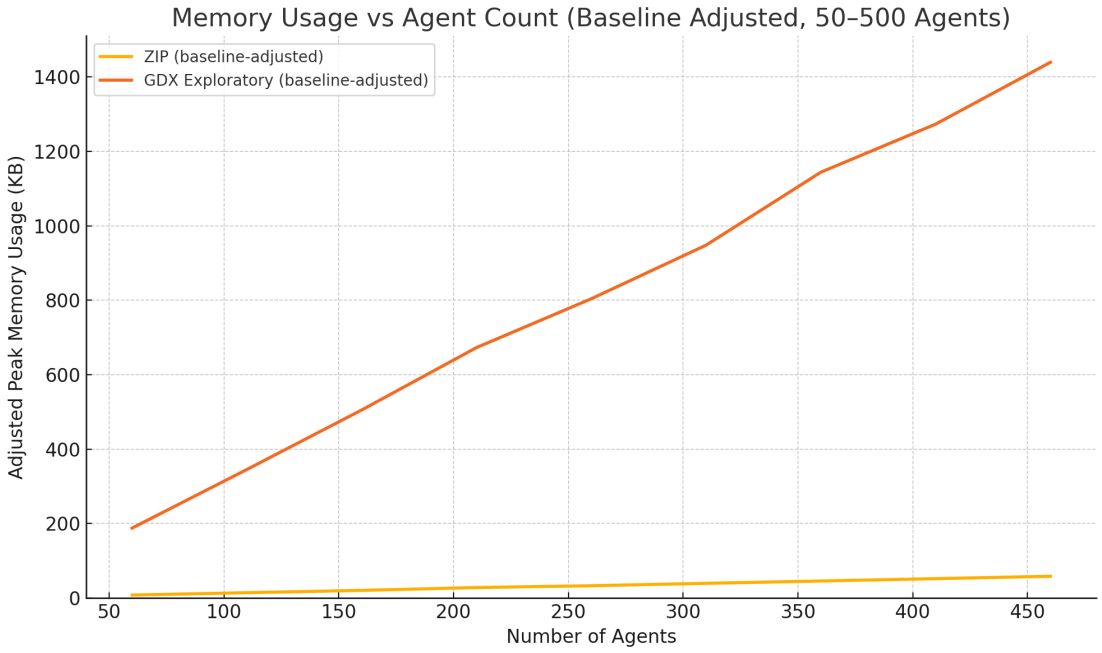

FIGURE 3.1: Memory usage comparison between ZIP and GDX

Given that our simulation bottleneck in the ABIDES framework is memory usage, primarily due to agent-exchange messaging and event logging (see discussion in Chapter 3.4.10), we chose to deploy the ZIP agent over GDX. This choice reduced memory overhead at the cost of slower price equilibration and was a trade-off made to run large-scale simulations.

3.1.3 Momentum

Momentum (Jegadeesh and Titman (1993)) is a technical trading style that determines the value of an asset based on the belief that asset prices have an element of inertia. These agents buy assets on an upward trajectory in expectation of a continuous rise in price and sell assets under the opposite conditions. A simple variant of the original strategy is shown below, where the agent chooses to buy the asset if the most recent o price observations exceed that of the most recent $o + \alpha$ observations and vice versa.

$$V = \begin{cases} q & \text{if } \sum_{t=1}^{o} p_{i-t}/o > \sum_{t=1}^{o+\alpha} p_{i-t}/(o+\alpha) + m \\ -q & \text{if } \sum_{t=1}^{o} p_{i-t}/o < \sum_{t=1}^{o+\alpha} p_{i-t}/(o+\alpha) - m \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Where V is the agent's purchase volume (with negative values indicating a sell order), q is an agent-specific parameter on their current purchase size, i is the current time, and m is the required deviation between two moving averages to warrant action.

3.1.4 Mean-Reversion

Mean-reversion (De Bondt and Thaler (1985)) is another technical strategy that contradicts the momentum strategy. It is based on an alternative finding where past losers significantly outperform past winners, due to the market participants' tendency to overreact to recent data. This leads to an undervaluation of the asset, making it a good buy in anticipation of a market correction. We also add simple mean-reversion agents in our simulation, which behave exactly in the opposite manner to the momentum agent.

$$V = \begin{cases} q & \text{if } \sum_{t=1}^{o} p_{i-t}/o < \sum_{t=1}^{o+\alpha} p_{i-t}/o + \alpha - m \\ -q & \text{if } \sum_{t=1}^{o} p_{i-t}/o > \sum_{t=1}^{o+\alpha} p_{i-t}/o + \alpha + m \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

3.1.5 Market-Maker

In general, market-makers are also profit-maximising agents who must balance conflicting priorities. They aim to maximise both profit margin and transaction volume, which push the bid-ask spread in opposite directions. They also seek to minimise the adverse effects of inventory exposure while maintaining competitive quotes. Designing a market-maker that performs well across these objectives is complex and remains an active area of research.

However, in our simulation, the purpose of the market-maker is not to generate profit but to guarantee a minimum level of market liquidity. To this end, we implement a relatively basic spread-based ladder strategy (Algorithm 3). Upon each wake-up (every second), the market-maker examines the current mid-price and maintains a fixed-width window around it. It populates both sides of the order book symmetrically by posting multiple price levels, up to a fixed number of ticks away from the mid-price. When the mid-price moves, it cancels orders that fall outside the new window and replaces them with new orders anchored relative to the updated mid-price. The agent does not observe trades or adjust based on execution history; it simply reacts to changes in the mid-price and posts fresh liquidity at deterministic levels.

Algorithm 3 Spread-Based Ladder Market-Maker Algorithm

Observe current market best bid B' and ask A'
Compute mid-price $M = \frac{B'+A'}{2}$
if mid-price has changed **then**
    Cancel all outstanding bid and ask orders
    Update internal bid/ask order queues relative to M
end if
for each price level i in bid ladder **do**
    Sample random order size q_i
    Submit buy limit order of size q_i at price $P_i = M - \delta_i$
end for
for each price level j in ask ladder **do**
    Sample random order size q_j
    Submit sell limit order of size q_j at price $P_j = M + \delta_j$
end for
Schedule next wake-up in 1 second

3.2 Desired stylised facts

Past studies have identified numerous stylised facts across asset classes in the financial markets. They show important properties of the market from an empirical perspective. These stylised facts fall largely under two categories: stylised facts regarding order size and market volume, and those regarding the change in price (often referred to as asset returns). In this section, we will discuss these stylised facts in more detail.

3.2.1 Asset returns

Vyetrenko et al. (2021) provides information on asset returns. The properties of asset returns were found to be consistent across multiple asset classes. A subset of their findings that we are interested in are shown below:

- Absence of returns autocorrelation for intervals larger than 20 minutes: $\text{corr}(r_t, r_{t+\rho}) \approx 0$
- Clustering of returns volatility: $\text{corr}(r_t^2, r_{t+1}^2) \approx 0$
- Heavy tails in the distribution of asset returns: $\text{Kurt}(R) > 3$
- Gain/Loss asymmetry, with higher volatility in losses: $\text{Var}(R >= 0) < \text{Var}(R < 0)$
- Correlation of traded volume and returns volatility: $corr(\sum_{t=i}^{i+n} v_t, \sum_{t=i}^{i+n} (r_t - \bar{r})^2/n)$
- Correlation of asset returns and returns volatility: $corr(r_n, \sum_{t=i}^{i+n} (r_t - \bar{r})^2/n)$

- Goodness of lognormal fit on number of orders in 5-minute intervals

Here r_t is the price at time t, v_t the volume, ρ a time interval greater than 20 minutes, and R the set of $r_{t+1} - r_t$ for all t.

3.2.2 Market volume

On the volume side, empirical findings and academic surveys by Abergel et al. (2016) show that the order arrival rate can be approximated using exponential or Weibull distributions. They also refer to studies that show that order sizes are distributed following a power-law. These studies show that the power-law distributions for limit order sizes have an exponent around 2.3 - 2.7 and market orders have an exponent around 2. They also indicate that orders tend to come in "round" sizes, such as 100s or 1000s. Vyetrenko et al. (2021) cites the literature that finds an intraday pattern on order volumes. It shows that volume peaks twice a day, upon market opening and near market close. They also indicate that a "U-shaped" polynomial can approximate this behaviour of trading volumes.

3.3 Cryptocurrency exchange

The rise in popularity of Bitcoin and other cryptocurrencies over the past decade has led to the emergence of numerous cryptocurrency exchanges worldwide, some of which facilitate daily trading volumes exceeding tens of billions of dollars. Unlike traditional stock exchanges, many of these crypto platforms provide open access to high-quality market data via public APIs, primarily to attract a broader user base. This accessibility is particularly advantageous for academic research, as it removes many of the data-related barriers that typically constrain empirical financial studies.

Before discussing our data collection methodology, it is important to acknowledge a structural challenge associated with cryptocurrency markets. Due to their relatively recent emergence and decentralised architecture, these markets remain highly fragmented across a multitude of exchanges, many of which are still maturing in terms of infrastructure and reliability. This contrasts with equity markets, where a handful of dominant exchanges capture the majority of global trading activity, making the overall market structure more consolidated and transparent. As a result, it is inherently more difficult to form a unified view of market activity in the cryptocurrency space. Despite this concern, we chose to proceed with data collection in the cryptocurrency space, as the increasing share of the market and the volume of trading concentrated in large exchanges suggest that the activity on a major platform can serve as a reasonable proxy of

broader market dynamics. This assumption is further supported by the growing presence of arbitrageurs, institutional traders, and high-frequency trading firms, who help enforce price efficiency across fragmented venues. To further enhance the above assumption, we collect data from Binance (the world's largest cryptocurrency exchange at the time of writing) on the most liquid currency pair available on the platform: the Bitcoin to US dollar (BTC/USD) currency pair.

A reasonable question that may be raised is why we chose to conduct a new empirical study on stylised facts, given the existence of prior literature in this area. The motivation lies in the fact that much of the existing research is based on data predating the 2017 Bitcoin boom (Bariviera (2017); Chu et al. (2015); Eross et al. (2019); Phillip et al. (2018)), a period before institutional interest in cryptoassets had materialised. This raises questions regarding whether these earlier findings remain representative of the current market environment, which has undergone significant structural changes since. To address this gap, we present an updated empirical analysis using data collected in 2021. Although subsequent studies, such as Bruzgė et al. (2023), have since extended the empirical literature to 2023, these were not available at the time our project was conducted (2020–2021). Moreover, in this process, our independent empirical analysis was able to add novel findings to this domain (Cho and Norman (2021)). To our knowledge, we were the first to report a positive correlation between returns and volatility in the Bitcoin market, as well as to document both a power-law distribution of order sizes and the pronounced clustering of orders around round-number transaction values. These updated stylised facts serve as important benchmarks for calibrating our simulation framework, and for future researchers relying on past empirical studies.

3.3.1 Data collection

We initially collected BTC/USD exchange data over 14 separate days in March and April 2021 to use in our study. The data were in the form of minutely LOB snapshots, which were queried from the live feed every second, and time series of orders across the entire time interval, which were called from a historic database using the API. The LOB snapshot is 100 layers deep on both sides of the mid-price, and the time series contains information on order ID, price, quantity, and time of the transaction, as well as whether the order was a buy or a sell and whether it was transacted at the best possible price. Using these data, we are able to generate data such as the mid-price, total volume, and order arrival rate to analyse the property of the market.

After tuning the simulations based on the stylised facts found in the above data, Binance opened up more of their API in late 2021, allowing users to query saved snapshots of the LOB directly from the historic database. This simplified the data collection process, as live code which scraped the point-in-time LOB was no longer necessary. This

greatly increased the number of data points available for the purpose of this investigation. However, this new feature was only offered on the BTC/USD perpetual futures exchange. While the cash and the futures markets are highly correlated and are likely to share most of the market properties with each other, we acknowledge the potential inconsistency this change in datafeed may have on our research. Despite this drawback, we believe that the benefits offered by a substantially larger data set exceed the drawbacks that may arise from inconsistencies between the cash and the futures markets. Therefore, while we remain wary of the potential discrepancies, the second component of our research into replicating market impact is conducted using data from the futures exchange.

3.3.2 Properties of the market data

We analyse the crypto market using the cross-asset stylised facts discussed in Section 3.2. Starting with the asset returns side, the returns in the BTC/USD market (Figure 3.2: top row, left image) are distributed normally, with Kurtosis (Fisher) = 3.19, indicating the presence of heavy tails similar or greater than that of the equity market. However, the autocorrelation of these returns is nearly random irrespective of the correlation interval (Figure 3.2: top row, right image), which is not in line with the traditional markets that exhibit positive autocorrelation for periods up to 20 minutes. On the other hand, there is evidence for volatility clustering, with positive but declining autocorrelation of squared returns over a longer correlation interval (Figure 3.2: second row, left image). Likewise, the market exhibits a strong correlation between asset volume and price volatility (Figure 3.2: second row, right image). However, the returns/volatility correlations conveyed a completely different story. Unlike the claim made by Vyetrenko et al. (2021) suggesting a negative correlation, the BTC/USD exchange showed a slight positive correlation (Figure 3.2: third row, right image). As for the gain/loss asymmetry, there is little to no evidence of asymmetry in our sample of the crypto market (Figure 3.2: third row, left image).

Looking at the volume side, we also observe some deviation from traditional stylised facts. The number of orders in a 5-minute interval fits nicely under a log-normal distribution (Figure 3.2: fourth row, left image), and order inter-arrival rate looks to be exponentially distributed (Figure 3.2: fourth row, right image). However, the evidence for higher volume around major market open/close is difficult to see (Figure 3.2: last row, left image). Lastly, although the large bulk of orders seem to be distributed following a power-law distribution, there are irregularities across the spectrum that raise questions on the suitability of the power-law (Figure 3.2: last row, right image). However, there was evidence for larger order sizes around a "round" number of assets, as exhibited by peaks at 0.001, 0.002 etc.

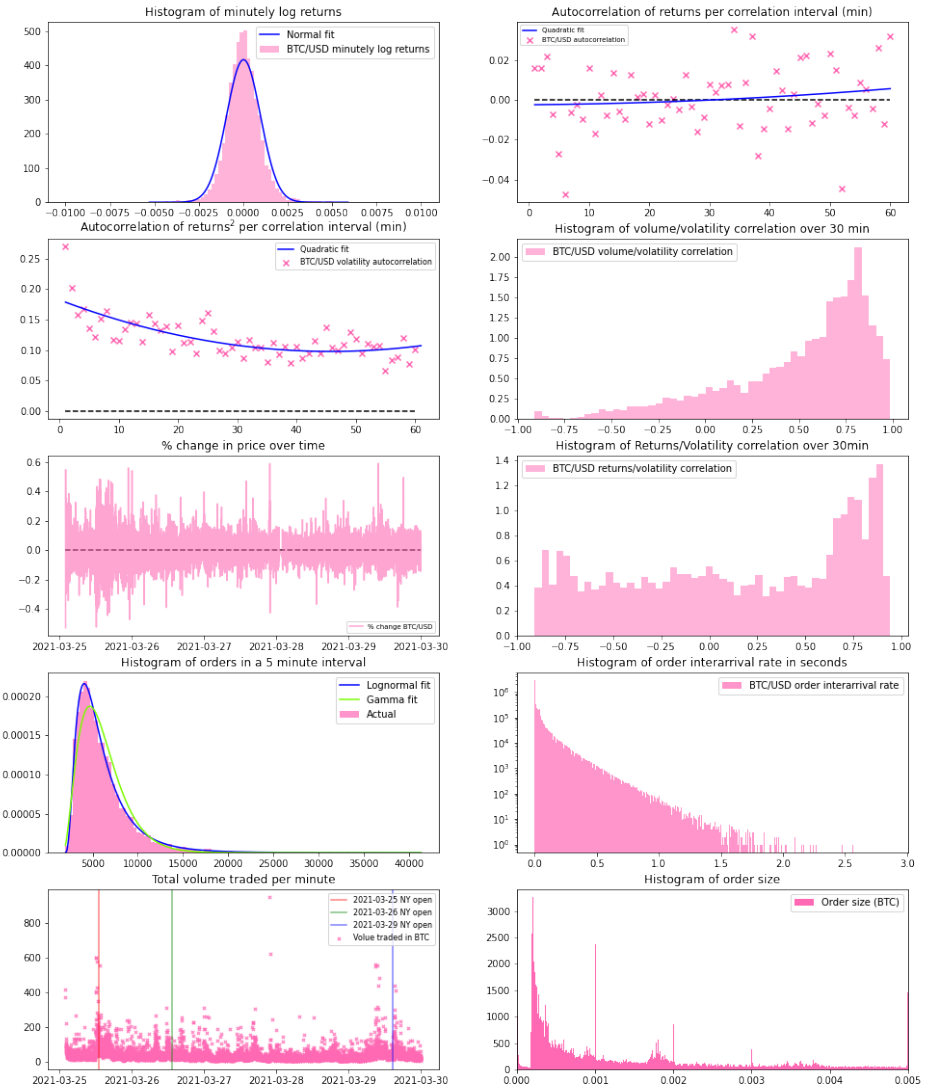

FIGURE 3.2: BTC/USD stylised facts (March 2021)

3.4 Tuning the simulation

In order to reach our goal of creating a financial exchange that is representative of the real world, the agents in our simulation have to be tuned against the collected data, such that they are more likely to exhibit the aforementioned properties. Within our agent-based simulation, several parameters can be altered to influence the market dynamics. Broadly, these parameters can be divided into micro and macro parameters.

The micro-parameters refer to rules and thresholds assigned to individual participants, including values such as minimum profit margins for a trade and the time between agent's orders. On the other hand, the macro parameters refer to a specific combination of agents that are being used to run the simulation. By tweaking the ratio of liquidity providers, technical traders, and fundamental traders, the overall market condition changes drastically.

We take a dual-pronged approach when tuning these parameters. To take advantage of the intuition and explainability offered by the agent-based approach at the individual participant level, the micro parameters are hand-tuned based on domain knowledge of the financial markets. On the other hand, the macro parameters are tuned using a computationally intense grid search, combined with estimations provided by utilising the response surface method.

3.4.1 Fixing trade volume

To reduce the dimensionality of our parameter optimisation, we fix the volume side of the agents' behaviour by forcing trade time and volume parameters. Whilst it may be possible to adjust agents' trading frequency or volume in a more natural manner by tuning agent-specific utility functions or adding more sophisticated trading conditions, we instead force every agent to follow the known stylised facts explicitly in order to speed up the simulation. This substantially reduces the number of dimensions that require tuning and reduces each agent's computational complexity, speeding up the simulation as a whole. To do this, we schedule our agents to submit orders based on a time interval sampled from an exponential distribution.

Since the exact parameter of the exponential distribution will vary depending on the asset and the number of participating agents, we decided only to match the distribution of order arrival, as opposed to matching the number of orders. This adjustment is necessary because the number of agents or arrival rate of agents will have to be very high in order to see a similar number of orders and transactions to the real exchange. In turn, we assume that varying the trading volume for all market participants by the same ratio will not affect other stylised facts in any other way. In a similar sense, the order size of each agent is forced to be drawn from a power-law distribution that is not fitted to the data. Lastly, to reflect the intraday pattern in our simulator, the arrival rate of agents is multiplied by a polynomial function of time from market open to market close. We do concede, however, that using this method may distort unspecified volume-related stylised facts such as volume/volatility correlation.

3.4.2 Initial results

Our initial hypothesis was to add a large number of fundamental traders to the simulation, alongside a smaller portion of technical and market-making agents, as we believe that there are more participants in the market making decisions based on something other than changes to asset price and volumes. Likewise, the proportion of market-makers was the smallest, as these types of participants are regulated and limited in many markets [1] in the real world.

The initial combination of agents with arbitrarily set individual parameters produces a reasonable set of results, with many desired properties of asset returns found in the simulation, albeit to varying extent. We present the result of our simulation in Figure 3.3 and Figure 3.4, and compare it with the actual BTC/USD market data from March 2021. Most stylised facts exhibit a similar property to that of the actual markets, but there are a few notable exceptions. Firstly, the correlation of volume and volatility (Figure 3.3: Second row, right image) is not nearly skewed enough. Although there is a clear skew in the simulated data, the crypto market shows a correlation that is substantially more pronounced, with a longer tail spanning in the negative direction. Secondly, the order sizes were not distributed in any way similar to the market data (Figure 3.3: Bottom row, right image). Unlike the market, the simulation shows that the most orders are placed with the lowest denomination of the asset. Lastly, the number of orders in a 5-minute interval (Figure 3.4: bottom left) does not fit the log-normal nor the gamma fit, which is in contrary to the actual market data shown above it.

Despite a few issues, this set of promising results gives us a good starting point to tune the simulation using the double-pronged approach we described earlier.

3.4.3 Manual tuning

The first step we take to improve our simulation is to adjust parameters that showed properties similar to those of the market, using only basic intuition.

The first area to tune is the wider variance of the log asset returns (Figure 3.4: top row, left). We believe that this is likely caused by the fundamental agents, which account for a large proportion of transactions, receiving a private valuation that is too noisy. This means that the agents have a vastly different private valuation of the asset, which can add volatility to the market as a whole. Although volatility could be coming from the aggressiveness of technical agents pushing prices away from the fair value in rapid succession, given their relatively small presence in the market, we believe the higher volatility is caused mainly by fundamental agents. Therefore, the fundamental agent's

[1] https://www.fca.org.uk/publication/documents/market-makers-authorised-primary-dealers.pdf, retrieved on 2021/05/07

FIGURE 3.3: BTC/USD stylised facts vs Untuned simulation results 1

observation error from querying the oracle for the agent's private valuation is reduced to address the discrepancy in market volatility between the simulation and the actual data.

Next, we look at the large discrepancy in the distribution of individual order sizes (Figure 3.4: bottom row, right). Unlike the real-world data, the order sizes in our simulation did not exhibit the expected power-law distribution. In an ideal world, we would adjust individual market participants' wealth and their willingness to buy an asset to replicate the empirical facts, as the agents' demand is determined by their ability and willingness to purchase an asset. However, it is difficult for us to differentiate our agents individually in this manner due to the complexity that it would add to the simulation. Therefore, we use an explicit function that generates agents' order size from a power-law distribution instead. A point to note here is that we have already included in our simulation the

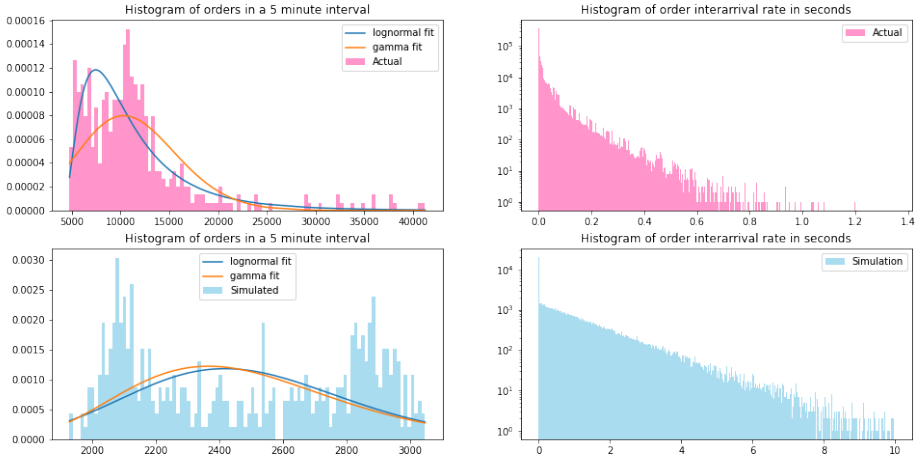

FIGURE 3.4: BTC/USD stylised facts vs Untuned simulation results 2

agents' tendency to prefer round order sizes, and this can be seen by the peaks in order sizes at $10 intervals.

Another striking difference between simulation and reality is the volatility in the volume traded (Figure 3.4: bottom row, left). The real-world data shows occasional spikes of traded volume over the trading day, with some evidence of increased volatility around the US market open. While this difference can be addressed by encouraging agents to participate more at certain times of the trading day (e.g., around major market opening hours) and linking their rate of market participation to levels of market volatility, we decide against this for two reasons. Firstly, the evidence for higher volatility around major market open/close is not strong, as exhibited by the lack of abnormal activity around London opening hours. Secondly, linking agents' participating frequency with market volatility adds a substantial computational load to our simulation. We believe that the computational resource we save by decoupling the frequency and volatility can be better used by increasing the number of participating agents in our simulation, which in turn makes the simulation more realistic.

Lastly, we want to point out the significant asymmetry in the number of orders in a 5-minute interval between the simulation and the real data. Although the orders in the current simulation (Figure 3.4: bottom left) do not look at all log-normally distributed, as sample size increases, we believe this will be resolved in a longer simulation. This is because the time between each consecutive order is exponentially distributed (Figure 3.4: bottom right) and as such, the distribution of orders per time interval will follow an inverse-exponential distribution. Sampling a large number of orders across multiple 5-minute intervals from this distribution will lead to a normal distribution of orders,

following the central limit theorem. However, because of the correlation between volume and volatility, this distribution will be positively skewed, producing a distribution similar to that of a log-normal distribution.

3.4.4 Manual tuning results

The intuition-based tuning of the micro-parameters yielded reasonable results. Using the newly tuned parameters, we ran a larger-scale simulation spanning 5 days, which saw most of the aforementioned differences between the simulation and the actual data disappear. Note that a longer 5-day period was used to reduce the likelihood of noise distorting our simulation results.

The simulation properties which looked broadly in line with the market in the initial simulation remained that way with the newly tuned parameters. With the increase in the number of data points from the longer simulation, we can now provide a clearer analysis of these market properties. Firstly, the decay in autocorrelation of price volatility became much more closely aligned with the actual data, and the decay itself exhibited a much smoother behaviour (Figure 3.6: second row, left graph). Secondly, the correlation between asset returns and asset volatility remained largely unskewed, and the simulation does well to replicate the uniform-like behaviour of the actual market correlation (Figure 3.6: third row, right graph). Lastly, the autocorrelation of the returns remained around the zero mark across all periods, and the anomalous point at the 1 minute mark that we saw in the initial simulation disappeared (Figure 3.6: top row, right graph).

A feature that yielded unclear results in the initial simulation due to the small sample size is the correlation between traded volume and return volatility. As the simulation duration increased, this feature became closer to the real data, displaying a negative skew between the two variables (Figure 3.5: second row, right graph). However, the gap between the data indicates that the simulation does not exhibit as strong a link between price volatility and traded volume. In other words, for the simulation to become more realistic, we require more agents to participate in periods of high price volatility and vice versa.

Likewise, the issue regarding the number of orders in a 5-minute interval also resolved itself to a respectable degree, as per our line of reasoning in the previous section. However, we note that the log-normal fit distribution does not fit our results very well (Figure 3.6: bottom left), as there is a clear presence of a long tail in the histogram. The long tail indicates that there are more periods of abnormally high volume within our simulation.

Looking at the features that we actively tuned from the initial simulation, we see a mixed set of results. The explicit inclusion of a power-law sampling for the order size yielded

a questionable result, with an element of power-law decay from the peak, but also an undesirable distribution of order sizes to the left of the peak (Figure 3.5: bottom row, right). An investigation into the issue revealed that this is the fault of the market simulator, which records orders in an undesirable fashion, where transactions are recorded from a limit-order perspective. This means that a market order of size 100 executed against three existing limit orders of size 45, 30, 25 is recorded as three small orders, as opposed to the single large order of size 100. As we can now guarantee that our orders are from a power-law distribution, the mismatch between the real data and our simulation shown in the histogram of order sizes will be ignored.

On the other hand, our tweaks to the observation error of fundamental agents reduced the simulation's price volatility to be almost exactly in line with the actual market data (Figure 3.6: top left). While we recognise that we could have adjusted the parameters of technical agents (profit margin, lookback period) to achieve the same result, we believe it is more effective to tune the fundamental agent's private valuation, as that value is derived exogenously to the simulation and therefore less likely to influence other stylised facts.

The above point about limiting the influence of parameter tuning to other stylised facts is an important one to stress. Both the long-tailed nature of the orders in a given interval and the difference in skewness of volume/volatility correlation can be fixed individually if we ignore the impact the changes bring on other market properties. However, maintaining the existing market properties (which are desirable) while fixing these properties proved difficult. In addition to this challenge, our agents are unable to vary their arrival rate or order size adaptively to the market, due to the additional computational complexity this will introduce to the simulation. This means that the tools we can use to make the simulation more realistic are also limited, making the issue more of a challenge. These constraints and the intertwined nature of stylised facts is the reason why we opt for a macro-parameter tuning methodology following the manual adjustments we discussed in this section.

3.4.5 Macro parameter tuning

To find the empirically optimal agent combination, we started out with the idea to search the combination space methodically using a grid search method. In particular, to avoid getting unlucky results from systematic problems, Latin Hypercube (McKay et al. (2000)) method was seen as the best candidate. This particular method guarantees a minimum number of random samples in a predetermined n-dimensional grid space, providing a better understanding of the simulation parameters.

FIGURE 3.5: BTC/USD stylised facts vs micro-parameter tuning results 1

However, for the purpose of our model, the total number of participating agents is kept constant, imposing a constraint on the macro-parameters. In addition, to keep the simulation run time at a reasonable level, the Zero-Intelligence Plus agent (ZIP) was capped at a maximum of 200. This is due to the substantially higher computational complexity of the ZIP agents when compared to other participating agents, thus they cost more compute time than the others. Adding such constraints renders the search space unrepresentable by evenly sized multi-dimensional grids, making standard grid search methods substantially more difficult to use. A solution to this would be finding a geometrically correct method to divide up the space equally without introducing sampling bias to maintain the benefits of a grid search. However, developing an algorithm to divide constrained space into unbiased sections is beyond the scope of this research and is left for future researchers to attempt. Instead, we opt for a simpler and more intuitive random search method, whilst remaining wary at the chance that some areas of the macro-parameter space may not be searched effectively, or in the worst case, at all.

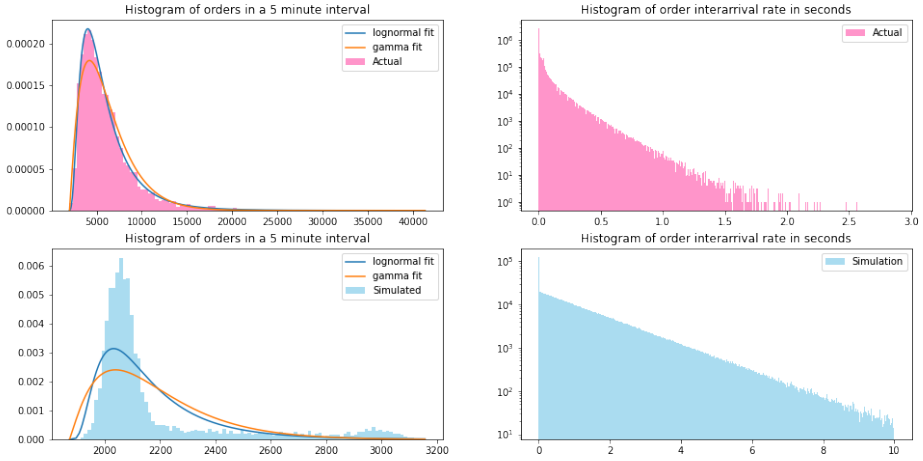

FIGURE 3.6: BTC/USD stylised facts vs micro-parameter tuning results 2

However, there is an advantage to the random search method over grid search. Unlike grid search, random search has the ability to add more simulation samples in small num

bers, as it is easy to sample a handful of additional configurations to add to the response surface, without introducing bias. This is in contrast to the grid search method, where additional simulation can only be added in multiples of the number of "grids". In turn, this translates into a more efficient use of computing resources, as each simulation we run is "embarrassingly parallel", thus able to benefit from using small amounts of free computational resources to add additional samples over time.

The random search we use is very rudimentary, but selects parameters in an unbiased manner. The search randomly chooses four numbers between 0 and 1000 representing ZI, ZIP, Mean-Reversion, and Momentum agents, respectively, and only proceeds to a simulation if the random numbers happen to sum up to 999 (we reserve a place for the market maker) and that there are a maximum of 200 Zero-Intelligence Plus agents. We repeat this search until we obtain the desired number of agent configurations to simulate. We are certain that there exists a substantially more efficient method of generating unbiased sample configurations, however given the relatively low cost of this rudimentary Monte-Carlo search, its ease of implementation, and the lack of this methodology's significance on our research objectives, we chose not to find a more efficient solution.

3.4.6 Macro tuning results

The initial search consisted of 942 simulations across the constrained space we described in the previous section. While many of the agent combinations yielded stable results, some gave results that diverged away from the historical price time series and quickly reached the computational limit (the computer's equivalence to positive and negative

infinity). Our initial suspicion was that the simulation was populated with a large number of technical agents that place diverging orders based on past market price, amplifying the noise made by fundamental participants. An investigation into these diverging simulations confirmed our suspicion that a large number of technical agents betting on the market was the cause behind this. We also discovered that the system reaches this unstable point when technical agents comprise about 40% of the market participants (Figure 3.7). Unfortunately, this has meant that many of our simulations give unreliable and misleading stylised facts due to the price diverging to computational limits, and therefore unable to derive much useful information out of this initial macro-search.

FIGURE 3.7: Simulated vs Actual midprice for varying number of technical agents

Following this finding regarding the stability of the simulation, we imposed a new constraint on the number of technical agents to encourage stable trading behaviour driven mostly by fundamental asset values. The new constraint limits the maximum number of each technical agent to 200. This new constrained space was sampled using the same random search algorithm, and we ran 497 simulations to estimate the response surface.

3.4.7 Response surface method

Market simulations with varying agent configurations produced a wide range of stylised facts. In order to use the result of these simulations in a macro-parameter search, we use

a response surface method to find the empirical optimum across all the stylised facts. As we are interested in the contribution of all four agents on numerous stylised facts that were identified in Section 3.2, our optimisation process comes in two steps. First, we fit a quadratic response surface for each stylised fact, sf_n, using least squares regression, where the number of each type of agent, x_i, is taken as input. The regression model for the surface is shown in Equation 3.6. A key point to note here is the exclusion of a linear term that represents one of the four types of agents. This is done to avoid introducing perfect multicollinearity arising from the total agent count constraint we impose on the system.

$$sf_n = \sum_{i=1}^{4}\sum_{j=1}^{4} a_{ij}x_i x_j + \sum_{i=1}^{3} b_i x_i + c + \epsilon \tag{3.6}$$

The second step is to manage the trade-offs between the various stylised facts, n, whilst optimising the agent configuration. We do this by using a cost function (Equation 3.7) that minimises the squared percentage difference between the real-world stylised fact, rw_n, and the stylised fact obtained by varying the agent configuration on the fitted response surface, sf_n. Note that the percentage difference is taken as a measure to standardise and equally weight every stylised fact of interest.

The optimisation routine yields a range of outcomes when the starting configuration of agents is changed. This indicates the abundance of local minima in our global response surface function. Nonetheless, our methodology is capable of producing reasonable agent configurations from the optimisation routine with locally optimal market behaviour.

$$\begin{aligned} &\text{minimize} && \sum_n ((sf_n - rw_n)/rw_n)^2 \\ &\text{subject to} && zi + zip + mr + mmt = 999 \end{aligned} \tag{3.7}$$

3.4.8 Tuned simulation result

Our final results come from a simulation that uses the manual adjustment performed in Section 3.4.3 and the configuration obtained from the random search in Section 3.4.5 as parameters. This simulation was run over the same 5-day period as the real-world baseline used in Section 3.4.3, and its results are shown in Figure 3.8 and Figure 3.9. For these plots, we chose one of many solutions obtained from Equation 3.7 that does not remove any of the participating agents. Specifically, this simulation contains 622 zero intelligence agents, 159 zero intelligence plus agents, 111 momentum agents, 107 mean reversion agents and 1 market-making agent[2].

[2]To enable reproducibility of our research, we have made our agents (designed to run in the ABIDES simulator) openly available to the research community: https://github.com/chris-jh-cho/bit_by_bit

We can immediately see from the simulation results that the vast majority of the stylised facts are in line with the market data. Apart from the order size histogram (Figure 3.8: Bottom row, right), which faces the aforementioned data collection difficulties linked with the design of the simulator, the only parameters that we could not replicate in our simulation are the skewness of volume/volatility correlation and the distribution of orders in 5-minute intervals (Figure 3.9: Bottom row, left). Outside of these, our simulation was able to successfully replicate the market behaviour on the distribution of asset returns, autocorrelation of returns and its volatility, gain/loss symmetry, and returns/volatility correlation.

FIGURE 3.8: BTC/USD stylised facts vs Final tuned results 1

3.4.9 Tuned simulation setup

The setup of our tuned simulation within the ABIDES platform is summarised in Table 3.1. A single BTC/USD simulation represents an eight-hour trading session. To

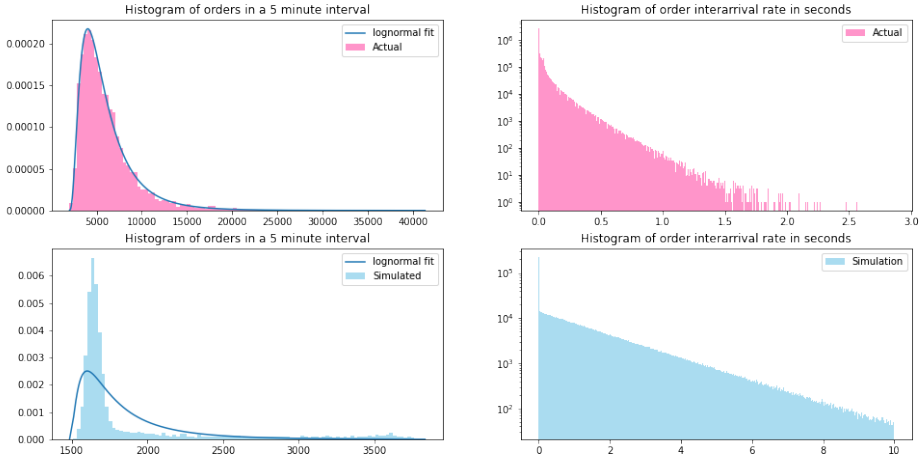

FIGURE 3.9: BTC/USD stylised facts vs Final tuned results 2

simplify financial accounting while maintaining realism, agents begin with a cash balance of zero and are permitted to accumulate negative balances. The smallest unit that can be traded (a "tick") is set to 0.00001 BTC.

Zero-Intelligence (ZI) agents are constrained to only submit orders that result in non-negative surplus, i.e., they do not trade unless the order is expected to be profitable. In contrast, Zero-Intelligence Plus (ZIP) agents dynamically adapt their profit margins via internal learning mechanism and are not subject to a hard surplus threshold. Similarly, the technical agents (momentum and mean-reversion) operate on the basis of observed price signals and execute trades when the signal deviates by a fixed amount ($1).

Non-market-making agents interact with the market according to a Poisson process, waking on average once per minute. In contrast, the market-making agent wakes every second, maintaining a continuous presence in the order book. This asymmetry in wake-up frequency reflects the respective roles of the agents. Market-makers actively provide liquidity to the market to stablise the price, but the rest of the participants can be seen as opportunistic.

Each simulation involves a heterogeneous mix of trading agents, including ZI, ZIP, momentum, mean-reversion, and a single market-making agent. The detailed logic for each agent class is described in Chapter 3.1. In the tuned configuration, the simulation includes 622 zero-intelligence agents, 159 zero-intelligence-plus agents, 111 momentum agents, 107 mean-reversion agents, and 1 market-maker agent.

Agents that require a notion of fundamental value query a simulated oracle, which returns the prevailing price based on historical data. At the start of each simulation, the oracle is seeded with a randomly selected 8-hour window of BTC/USD midprice data

from March and April 2021, sourced from Binance. To introduce dispersed private valuations while preserving information efficiency, the oracle adds zero-mean Gaussian noise with standard deviation $2.12 to the true price. This heterogeneity leads to trading activity among agents with otherwise symmetric rules. In contrast, technical agents do not consult the oracle; they base their trading decisions on short- and long-term moving averages of the last 15 and 30 trades, respectively, to identify momentum or reversion opportunities.

3.4.10 Computational considerations

All simulations were executed on the University of Southampton's Iridis 5 high-performance computing (HPC) facility. Each compute node provides 40 CPU cores running at 2.0 GHz, with 192 GB of shared memory. This places an effective upper bound of 4.8 GB of RAM per core if the full throughput of the node is to be utilised efficiently.

In the original ABIDES configuration, simulating 1,000 agents arriving on average once per minute —drawn from a mixture of ZI, ZIP, GDX, momentum, mean-reversion, and market-making agents —took between 8 and 12 hours to complete and consumed around 30 GB of RAM per run. In practice, memory was the limiting factor. Although many cores were available, the high memory usage per simulation significantly restricted the number of concurrent jobs that could be run on a single node.

To mitigate this bottleneck, several memory-saving decisions were made. First, we represent fundamental agents using the ZIP algorithm rather than GDX, which retain adaptive pricing behaviour but require significantly less memory and compute overhead. Second, we disabled logging features that were not required for our analysis, such as order latency tracking and detailed transaction message traces. Third, we fixed agent micro-parameters and used homogeneous agent populations for each agent type during batch simulations, which further improved runtime and helped isolate the effects of macro-level configuration changes.

Following these optimisations, a single simulation run could be completed in 3 to 4 hours, using approximately 20 GB of RAM. While this still does not maximise the throughput of the node, this increased the size of simulation batches in parallel, allowing the experiments to be conducted within the operational constraints of the HPC facility in a more reasonable time frame.

It is important to note that these adjustments were made to enable large-scale exploration, not as shortcuts. Our methodology requires extensive simulation sweeps to conduct macro-level parameter tuning. Once optimal configurations were identified, individual runs could be executed efficiently, even on a modern desktop machine. Without the need to log every aspect of market activity for stylised fact calibration, the memory footprint drops to just 3–4 GB. Thus, by identifying an "optimal" agent configuration to

TABLE 3.1: Tuned simulation configuration

Component	Value / Description
Base ABIDES simulation configuration	
Simulation duration	8 hours (continuous trading session)
Oracle data source	Historic BTC/USD midprice from Binance
Oracle data window	Random 8-hour slice from March and April 2021
Tick size (minimum traded size)	0.00001 BTC
Communication latency	0 (agents operate with zero message delay)
Initial cash	0 (negative balances allowed)
Market-maker (1 agent)	
Wake-up frequency	Every 1 second
Quoting strategy	Submit symmetric ladder of limit orders around mid-price
Number of ticks	5 price levels per side
Orders per level	5 per price level
Order refreshing	Cancel all and re-submit if mid-price changes
Momentum (111 agents) and Mean-Reversion (107 agents)	
Wake-up frequency	Poisson: avg. once every 60 seconds
Order size	$\lceil 70/U \rceil$, $U \sim \text{Beta}(3.5,\ 1)$; rounded to nearest 10 with 20% probability
Short-term signal window	15 most recent trades
Long-term signal window	30 most recent trades
Trigger threshold	Execute trade if price difference $\geq$ \$1
Zero-Intelligence (622 agents)	
Wake-up frequency	Poisson: avg. once every 60 seconds
Order size sampling	$\lceil 70/U \rceil$, $U \sim \text{Beta}(3.5,\ 1)$; rounded to nearest 10 with 20% probability
Minimum surplus constraint	None (trade has to be profitable)
Observation noise	Gaussian, standard deviation = \$2.12
Zero-Intelligence Plus (159 agents)	
Wake-up frequency	Poisson: avg. once every 60 seconds
Order size sampling	$\lceil 70/U \rceil$, $U \sim \text{Beta}(3.5,\ 1)$; rounded to nearest 10 with 20% probability
Observation noise	Gaussian, standard deviation = \$2.12
Learning rate (β)	Uniform $[0.1,\ 0.5]$
Momentum term (μ)	Uniform $[0.2,\ 0.8]$
Initial profit margin	Uniform $[0.3,\ 0.35]$
Absolute change (up)	Uniform $[0.0001,\ 0.001]$
Relative change (up)	Uniform $[1.00,\ 1.05]$
Absolute change (down)	Uniform $[-0.001,\ -0.0001]$
Relative change (down)	Uniform $[0.95,\ 1.00]$

reproduce stylised facts on the ABIDES platform, our work enables future researchers to run simulations that are representative of the market without requiring access to supercomputing resources.

3.5 Conclusion

In this chapter, we have developed and validated a novel simulation framework designed to replicate the microstructural dynamics of the Bitcoin exchange using the ABIDES platform. Our agent-based methodology combines hand-tuned micro-parameters and response surface guided macro-parameter tuning to match a comprehensive set of stylised facts derived from market data we collected from Binance. This differentiates itself from many existing agent-based simulation practices, which often look at realism through behavioural perspective over empirical alignment with real-world market data.

We began by defining a simplified taxonomy of market participants - fundamental, technical, and liquidity-providing agents - and implemented representative agent behaviours through established algorithms (ZI, ZIP, momentum, and mean-reversion). Whilst these models are simple by design, they retain sufficient heterogeneity to support the efficacy and robustness of the simulation.

Empirically, we first revisited and extended the literature on stylised facts in cryptocurrency markets using a fresh dataset from 2021, addressing our first research objective. Our findings confirmed and supplemented previous results, including volatility clustering, power-law order size distributions, and round-number effects in trade sizes. These findings then formed the target values for our simulation's synthetic market microstructure.

Our two-stage calibration process successfully adjusted the simulation to replicate key empirical regularities, thereby fulfilling our second research objective. Manual tuning of micro-level agent parameters aligned return distributions and volatility properties with observed market data, whilst automated macro-parameter optimisation refined the mixture of agent types to reproduce higher-order dependencies. Despite limitations in modelling adaptive behaviour and intraday periodicity, the final configuration captured a broad suite of empirical features.

This simulation contributes the first publicly documented agent-based model of a cryptocurrency exchange explicitly calibrated to match contemporary stylised facts. It provides a robust empirical testbed for evaluating market behaviour under controlled interventions and serves as the foundation for the reinforcement learning experiments to be discussed in subsequent chapters.

Future work could proceed along several directions. First, incorporating more adaptive agent behaviours could enable a richer examination of endogenous strategy development and market co-evolution. Second, extending the simulation to cover multi-venue environments would allow for the study of cross-exchange arbitrage and fragmentation, a key feature of cryptocurrency markets. Third, simulating market stress scenarios (e.g., flash crashes, liquidity droughts) could offer insight into systemic vulnerabilities

within the crypto market. Finally, whilst our focus was on stylised facts in the BTC/USD pair, replicating this methodology across other cryptoassets such as Ethereum, Solana or Doge would test its generality and extend its applicability to broader crypto market research.

Chapter 4

Incorporating Market Impact

In Chapter 3, our focus was on identifying stylised facts that could be observed passively in the market. After tuning the simulation to reproduce empirical observations, we presented a representative setup using one of the optimal configurations in Section 3.4.8. However, while this was an important step toward ensuring realism in our simulation, it only captured the passive characteristics of the market.

In this chapter, we shift our focus to an equally important component of realism: market impact. Specifically, we examine how the simulated market responds to the execution of orders. For the simulation to be credible, external agents' orders must be absorbed and executed in a manner consistent with real-world market behaviour. At the same time, unless an agent executes very large volumes, its actions should not distort the broader market direction or stylised facts. To this end, we make two key contributions. First, we present novel empirical findings on the decay of market impact and the autocorrelation of order signs over time, and we offer updated evidence for the *square root law* of market impact using post-2017 data. Second, we use these insights to develop a new simulation framework (PRIME) that can reproduce market impact whilst retaining price-tracking ability by using the same fundamental and technical agents introduced in Chapter 3. These contributions correspond specifically to the first and the third research objectives we outlined in Section 1.2 of this thesis.

4.1 Defining Market Impact

market impact refers to the change in the price of an asset caused by the trading activity on that asset. From the perspective of an individual order, the impact is largely deterministic, driven by the order size, type, and prevailing liquidity in the limit order book at the time of execution. This means that larger trades or thinner liquidity conditions result in greater price movement, or a higher market impact. Furthermore, liquidity

tends to increase linearly as price levels moves away from the best bid and ask prices, up to a certain distance where the LOB become sparse. This structure leads to a phenomenon in which increasingly larger volumes are required to move the price linearly away from the mid-price. This relationship is what underpins the well-known *square-root law* of market impact (Torre (1997)), which posits that the price impact of a trade grows proportional to the square root of the trade size.

4.1.1 Shape of the Limit Order Book

We start our analysis of market impact by first looking at the shape of the LOB in the BTC/USD Futures market. As discussed briefly above, the expected shape of the LOB is a linearly increasing one, with the volume of available limit orders increasing as the price moves deeper into the book (further from the mid-price). Looking at the average shape of the LOB from the real markets, we do not observe this on the top 20 levels of the LOB (Figure 4.1). Instead, there is a high concentration of order volume on the top level, which drops off sharply from the next level in, with no evidence of linearly increasing volume as price moves away from the midprice. However, evidence from previous studies on the BTC/USD market suggests that the top level indeed has a disproportionately large volume, and the LOB does increase linearly for the price range of up to 2% around the mid-price (Donier and Bouchaud (2016), Figure 6). Unfortunately, we cannot verify this information ourselves, as the LOB snapshots from Binance only show limited depth (top 20 levels, which correspond to 0.14% price width), nowhere near the 20% price range that was collected by Donier and Bouchaud (2016). Therefore, it is very difficult for us to make verifiable claims based on the shape of LOB. In addition, while the shape of the LOB is a useful observation to start with, it merely provides an overview of the fundamental mechanics of the limit order book, and not any additional insight into how the market reacts and replenishes itself ex-post transactions. To understand this, the LOB needs to be observed over a time interval.

4.1.2 Temporal Market Impact

The market impact becomes less mechanical when approached from a non-instantaneous perspective. This is due to the concept of impact reversion and hidden liquidity arising from what is commonly known as the *Latent Order Book*. This is the idea that most market participants do not place limit orders at their private valuation until their valuation becomes (or close to) the most competitive price in the book in order to increase the likelihood of the transaction. This strategy allows participants to withhold private information from the market, allowing them to better utilise their information advantage. For example, an individual willing to sell BTC at USD 100 may not place their limit order when the best ask price in the market is at USD 98. However, when a large

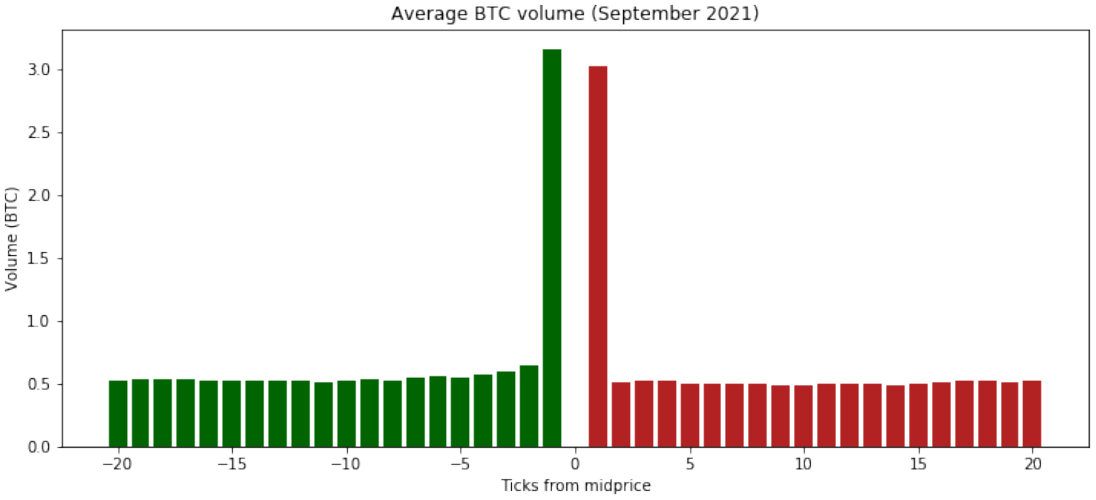

FIGURE 4.1: Average BTC/USD LOB Volume (Actual Data, September 2021)

buy order eats up all the liquidity between USD 98 and USD 100, pushing up the best ask price to USD 101, this individual is more likely place a limit order asking for USD 100. This behaviour is encouraged, as placing a large limit order at USD 100 when the market is at USD 98 would reveal to the market that a large seller is in play, thus influencing participants' decisions (e.g. would-be buyers may be more inclined to wait until the price falls, rather than transacting immediately). Consequently, it is only when the price level changes that the hidden liquidity is brought out into the market and causes a reversion in market impact over time. The *Propagator model* (Gatheral (2010)) is a widely used model that explains this phenomenon by dividing up the market impact into initial impacts and the reversion of these market impacts when estimating the total market impact over a set period. We too will use this framework presented by Gatheral *et al.* for the purpose of modelling impact in our research. Note that modelling the idea of latent liquidity in a fully consistent model remains an active area of research (Donier et al. (2015)), and is beyond the scope of our project. Instead, our work focuses on replicating the empirical outcomes of the simulated markets to be in line with the actual and exchange, from the lens of the *Propagator model*, and does not provide a novel mathematical explanation for our observations.

4.2 BTC/USD Futures

4.2.1 Initial Impact

As discussed above in Section 4.1, the market impact should be approached from a temporal perspective to avoid drawing conclusions based purely on the mechanical nature of the LOB (e.g., market order of size 100 consuming 100 units of limit orders, pushing

the price by 4 ricks). Therefore, we resample the trade data into 5-second time buckets, where the traded volume is defined as the net buy/sell volume across the interval (e.g. 200 buy orders and 500 sell orders within the interval will register as -300 orders), and price change is the difference between the close and open mid-price of the 5-second interval.

Looking at the resampled data from the BTC/USD Futures market (Figure 4.2), a positive relationship between trade size and price change can be observed. However, despite the generally positive relationship, some large discrepancies in impact between orders of similar sizes are observed. This is likely due to the differences in prevailing market liquidity and volatility mentioned previously. A common model to incorporate the prevailing liquidity, current volatility, as well as the nature of non-instantaneous market impact is the aforementioned *square root law* (Torre (1997)). The general form of this model is shown in Equation 4.1, where the market impact I is modelled as a function of the volume of the orders, Q, in the present time period, T, the "recent" time horizon (we use 1 hour). Here, σ_T is the average volatility over time T, V_T is the average traded volume over time T, δ the exponent of the diminishing market impact, usually assumed to be around 0.5 (hence the *square root law*), and a constant k.

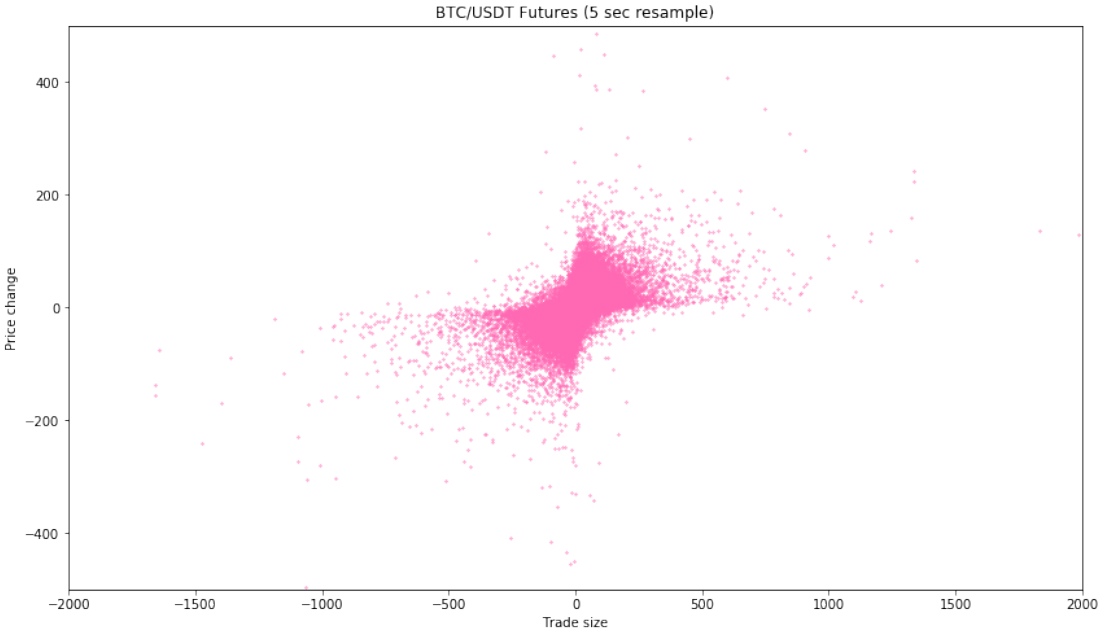

FIGURE 4.2: BTC/USD Market Impact (5 sec resample)

$$I(Q,T) = k\sigma_T\left(\frac{Q}{V_T}\right)^{\delta} \tag{4.1}$$

Assuming the exponent δ to be equal to 1 for the time being (i.e. no diminishing market impact), adjusting market impact by prevailing price volatility, σ_T, and the volume, V_T, yields plots that appear better correlated (Figure 4.3). The difference between the left

and right plots in this figure is the way in which average volume has been calculated, with the left plot using an arithmetic moving average, and the right using a triangular moving average with the peak of the weight placed at most recent time period.

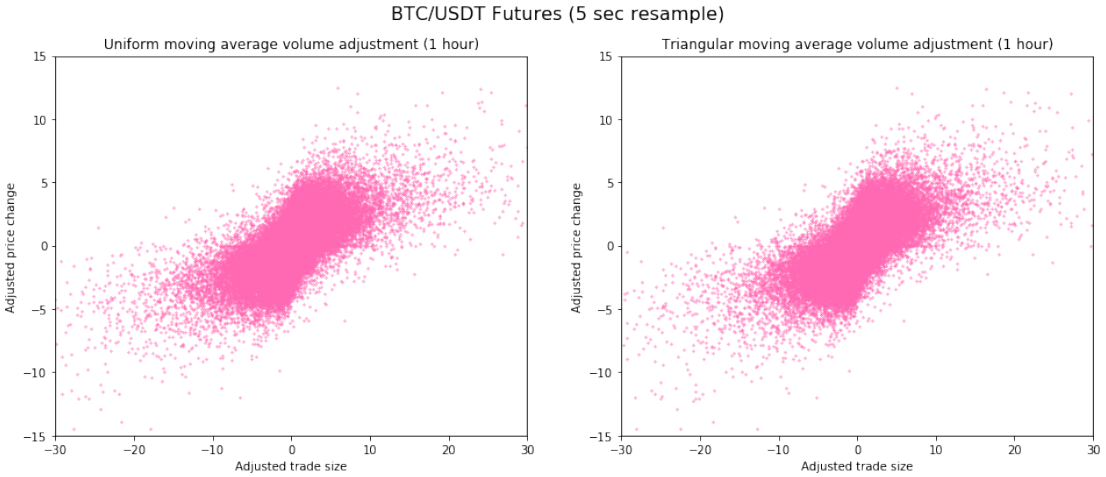

FIGURE 4.3: BTC/USD Market Impact (5 sec resample, volume & volatility adjusted)

This visually improved result following volatility and liquidity adjustment leads to an enhanced explanatory power of order size on market impact, when compared to the pre-adjusted data. However, we find this visualisation to be slightly misleading, due to the high density of data points across the board (there are over 800,000 data points in the graph, after plotting only 1% of the overall data). Instead, when the orders are gathered into buckets of order size and their mean market impacts plotted on top of the scatter plot (cyan lines, Figure 4.4), we find the relationship between order size and price change to be non-linear. Zooming into the bucketed means (Figure 4.5), increasing resistance of price change to traded volume can be observed much more clearly, suggesting that the exponent δ in Equation 4.1 indeed has a value less than 1.

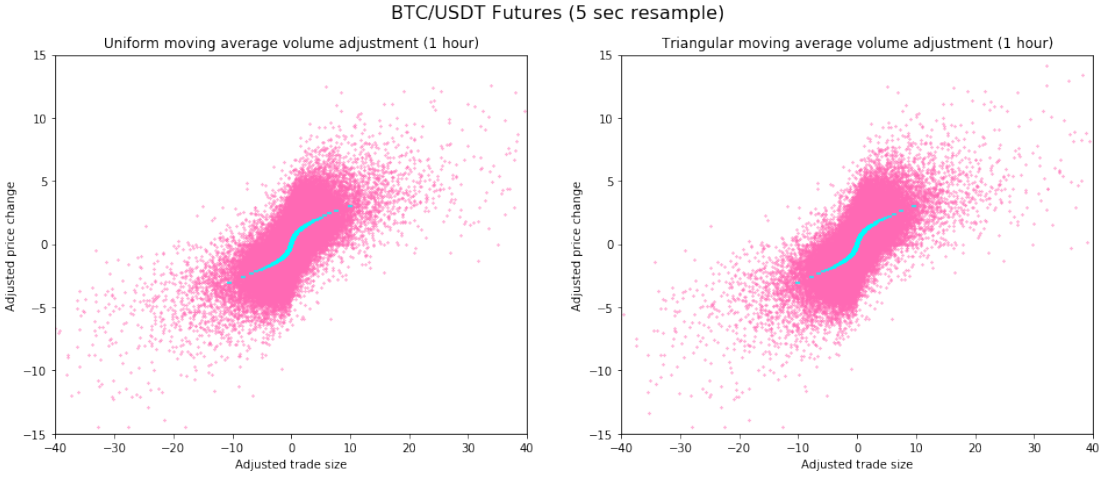

FIGURE 4.4: BTC/USD Market Impact with percentile bucket average impact (5 sec resample, volume & volatility adjusted)

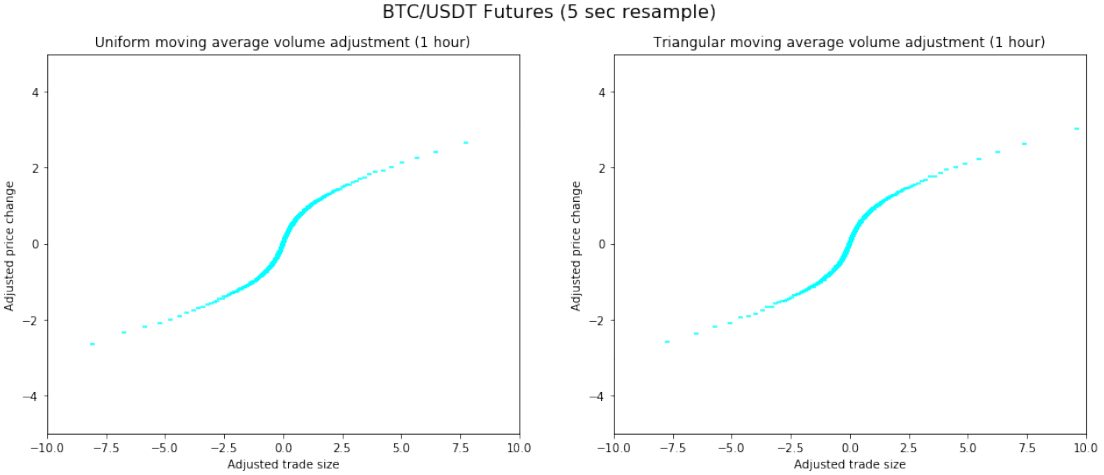

FIGURE 4.5: BTC/USD order percentile bucket average impact (5 sec resample, volume & volatility adjusted)

Using the data, we calculate the value of δ using an optimisation routine. The result of the optimisation yields the outcome $\delta = 0.58$. When this exponent is applied to the data following Equation 4.1, the bucketed mean impact is transformed to that shown in Figure 4.6. While explainability near zero trade size becomes non-linear, the relationship between trade size and market impact becomes linear elsewhere. This is the expected behaviour for two reasons. Firstly, small net orders tend to be traded in a differing liquidity environment on either side of the book. In other words, 100 units of buy orders followed by 100 units of sell orders will not necessarily bring the price back to the starting point, depending on the prevailing liquidity on the buy and sell side of the book. Second, the reversion of market impact from previous trades will account for a larger portion of the overall market impact near the zero mark (since the current market impact is small), distorting the relationship between current order size and market impact.

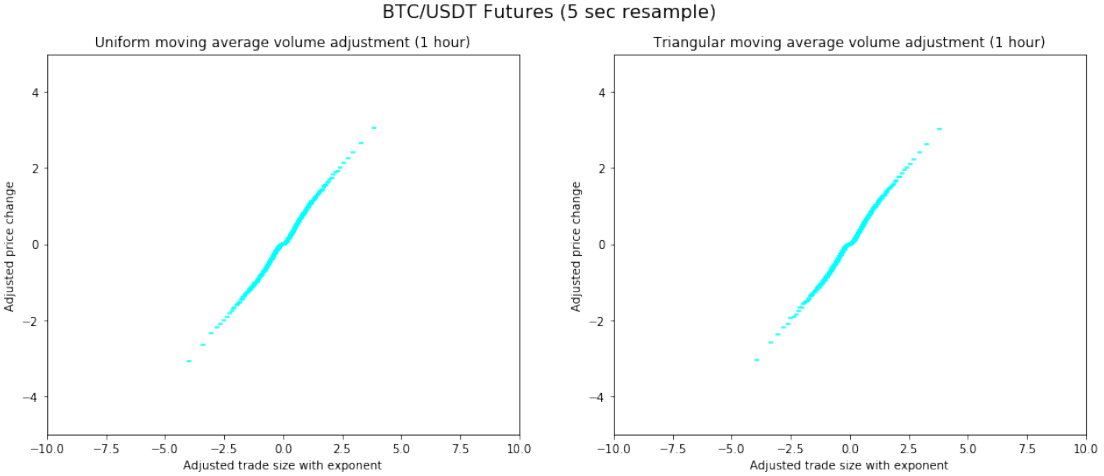

FIGURE 4.6: BTC/USD order percentile bucket average impact (5 sec resample, volume & volatility adjusted, including exponent)

Applying the δ exponent to adjusted trade size data results in a new scatter plot shown in Figure 4.7. Although the plot exhibits a slight kink around zero net volume due to the non-linearity issue discussed above, it nonetheless displays the relationship between trade size and price change explained by Equation 4.1 well. In addition, it provides empirical evidence of the initial market impact in the BTC/USD Futures market that can now be used as a baseline for our simulation.

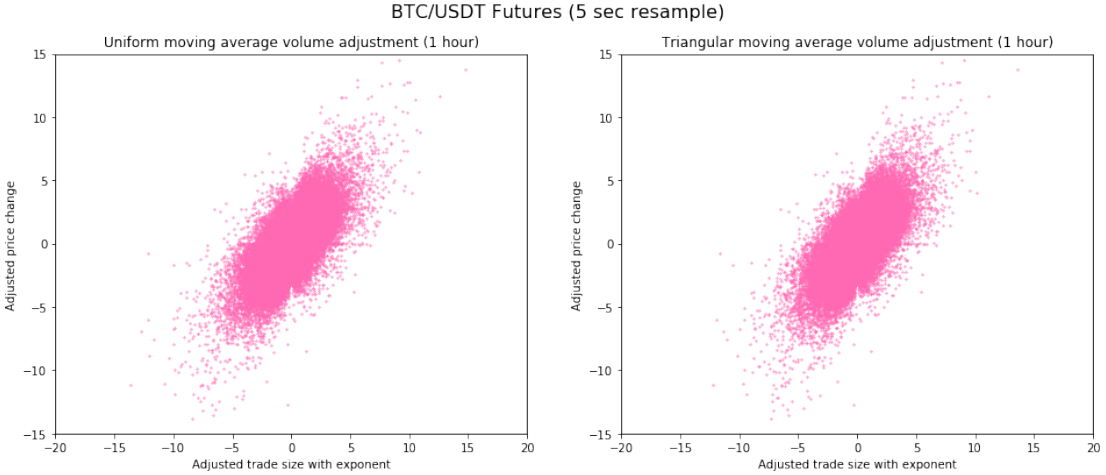

FIGURE 4.7: BTC/USD Market Impact (5 sec resample, volume & volatility adjusted with exponent)

4.2.2 Impact Reversion

We now present the evidence of the reversion in market impact that we have references multiple times in sections above. We start with the idea that if the reversion of market impact does not exist, there should be no intertemporal conditionality between trade size and market impact. Therefore, under this assumption, the sign of the previous period volume (net buy or net sell) should not influence the current market impact in any way. However, when the current market impact is divided into two groups based on the buy/sell status of the previous time period, we observe a gap between the averages of the two groups (Figure 4.8). This shows that the buy/sell direction of the previous period has an influence on the market impact of the current time period. In addition, as the name "reversion" suggests, if the previous period was a net buy, then the current period's impact is pushed in a negative (sell) direction vice versa if the previous period was a net sell.

As described by the *Propagator model* (Gatheral (2010)), we expect the price impact of a trade to revert in a decaying fashion, as the time between the trade and the observation increases. This follows Equation 4.2, where the total market impact, $M(t)$, over time period t is the accumulation of all initial impacts made by trades across the time period, $f(\dot{x}_s)$, after the decay of these impacts have been taken into consideration following the

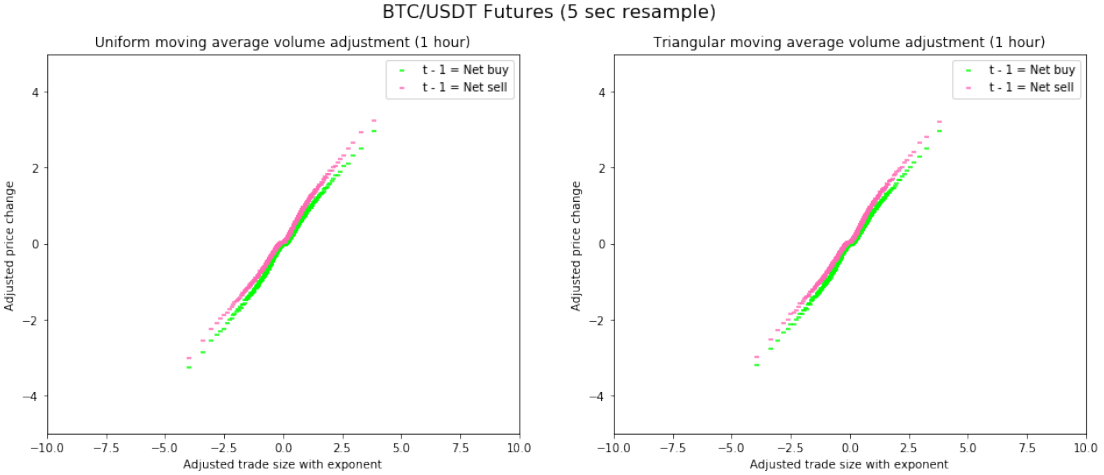

FIGURE 4.8: BTC/USD order percentile bucket average impact (5 sec resample, volume & volatility adjusted, including exponent)

decay kernel $G(t-s)$, plus a noise term δ. Here, the $\dot{x_s}$ represents the rate of trading across our time period t and is assumed to be constant (thus we assume trades are made at a constant rate, and the integral of $\dot{x_s}$ over the time period t is equal to the net value of trades across this period). We have shown in the previous section that the initial impact follows the *square root law* after adjustments. Therefore we can calculate the initial impact using Equation 4.1, substitute $f(\dot{x_s})$ with the value, and use this information to find out the functional form of the decay, $G(t-s)$, in the BTC/USD Futures market in an empirical manner. Note that we ignore the noise term δ and assume that it is zero in our work.

$$M(t) = \int_0^t f(\dot{x_s})G(t-s)\,ds + \int_0^t \delta\,dZ_s \tag{4.2}$$

By regressing the adjusted order size from time $t-100$ to t against adjusted price change at time t for all available t, the resulting coefficients show the average market impact of a unit trade on the present and future periods (Figure 4.9). The coefficients show a large initial impact at $t=0$, followed by a reversion in impact, as exhibited by small negative coefficients at $t>0$. When the cumulative market impact of a unit order at time t is plotted over 100 periods as before (Figure 4.10), a decay in market impact following a power law can be observed. This evidence alongside our argument presented in Figure 4.8 clearly shows the presence of reversion in market impact within the BTC/USD market.

4.2.3 Meta Orders

The final section of our analysis of the BTC/USD Futures market is regarding meta orders. In general, large orders are known to be broken down into many smaller orders

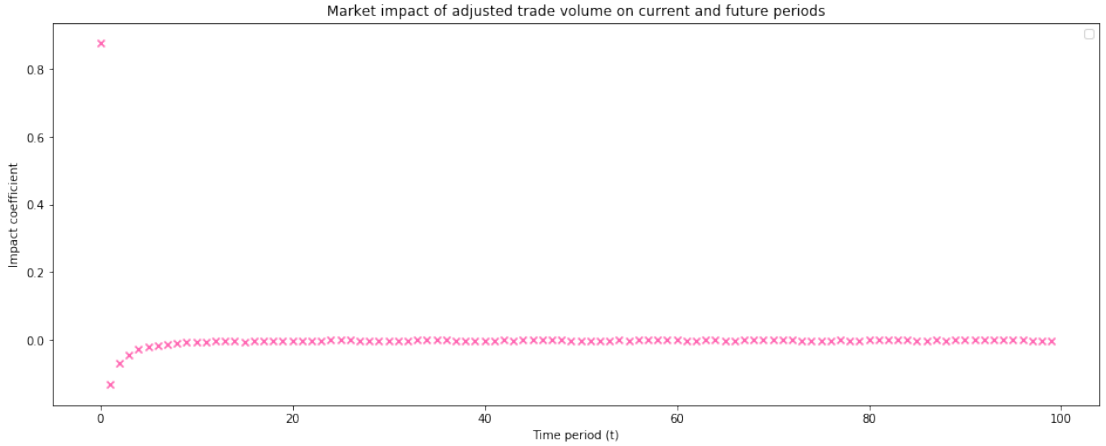

FIGURE 4.9: BTC/USD market impact coefficient over time (5 sec resample, volume & volatility adjusted, including exponent)

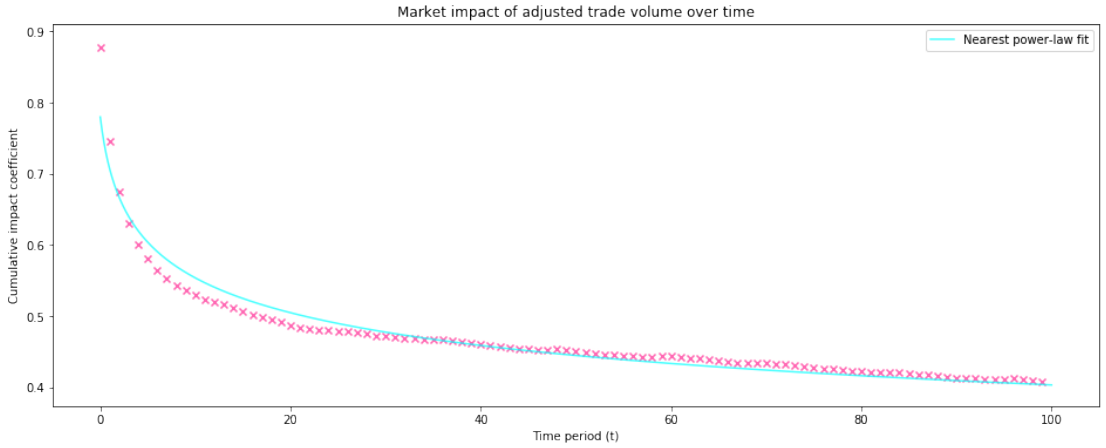

FIGURE 4.10: BTC/USD market impact decay over time (5 sec resample, volume & volatility adjusted, including exponent)

in an attempt to hide the total transaction volume and reduce the market slippage of the transaction. This is done using various methods, including the well-known Volume Weighted Average Price (VWAP) and Time Weighted Average Price (TWAP) execution strategies. Breaking down a large order into smaller chunks leads to the idea of meta orders, which spawn consecutive orders in the same direction. This idea can be verified by checking the autocorrelation of the order signs between consecutive orders.

The evidence behind this autocorrelation is covered in detail in Bouchaud (2018), Section 2.8, and the decay in autocorrelation over longer lags is shown to be well approximated by a power law ($l^{-\gamma}$ with $\gamma < 1$) across traditional asset classes. Our investigation into the data collected from the BTC/USD exchange reveals that the direction of the trade sign does exhibit long range autocorrelation even in the Cryptocurrency space, which also decays following a power-law (Figure 4.11). This autocorrelation is also a feature we choose to include in our market simulation, in an attempt to mimic the transaction dynamics added by meta orders in the market.

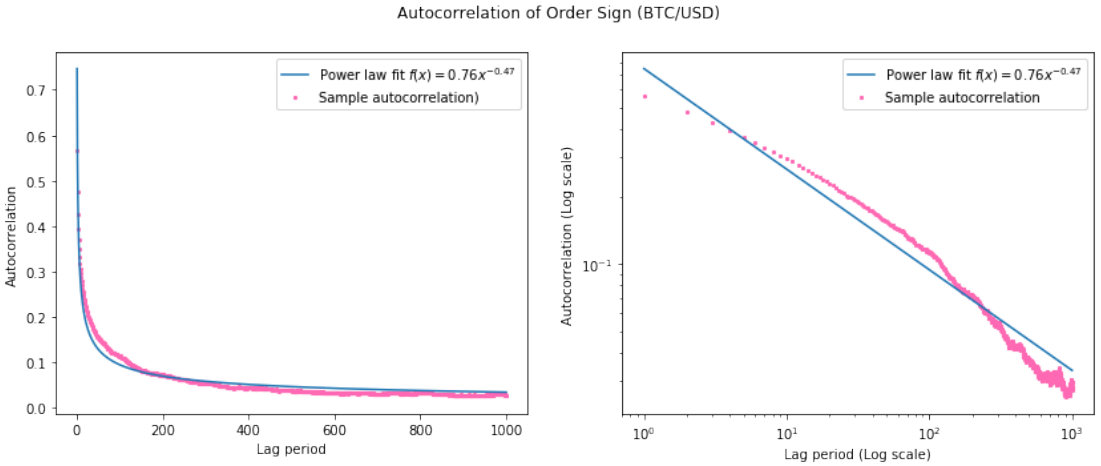

FIGURE 4.11: BTC/USD Autocorrelation of Order Sign (Actual Data, March 2021)

4.3 Baseline Simulation

In section 4.2, we provided empirical evidence for market impact observed in the BTC/USD Futures exchange using the *Propagator model*. This allowed us to divide the impact into two components: the initial impact and the decay kernel. On top of this, we explored the idea of meta-orders towards the end of our analysis. From this section onwards, we show how we replicated these observations in our simulation framework. As a starting point, we analyse and assess our model of the LOB we developed in Section 3.4 using the same framework as in Section 4.2. We refer to this model from the previous chapter as the *Baseline* model henceforth.

4.3.1 Initial Impact

As first step, we look at the initial impact observed in this simulation. Before incorporating any volume or volatility adjustments, the relationship between trade size and price change shows a positive correlation (Figure 4.12) but exhibited a very different behaviour from that found in the real world seen earlier (Figure 4.2). In fact, a very linear market impact is observed in the simulation result, prior to applying any adjustments via volume, volatility, or the exponent.

Upon adding adjustments for prevailing liquidity and volatility, the scatter plot appears to have improved. The small number of periods we observe in Figure 4.12 that have very small net trade size (in absolute terms), but large changes in price all but disappear after the adjustment, offering a much "cleaner" looking cloud of data points (Figure 4.13). However, the market impact remains linear, and does not show the diminishing behaviour that we observed in the real world (Figure 4.5). This linear relationship we observe in the *Baseline* model is perhaps unsurprising, as the agents in this model were

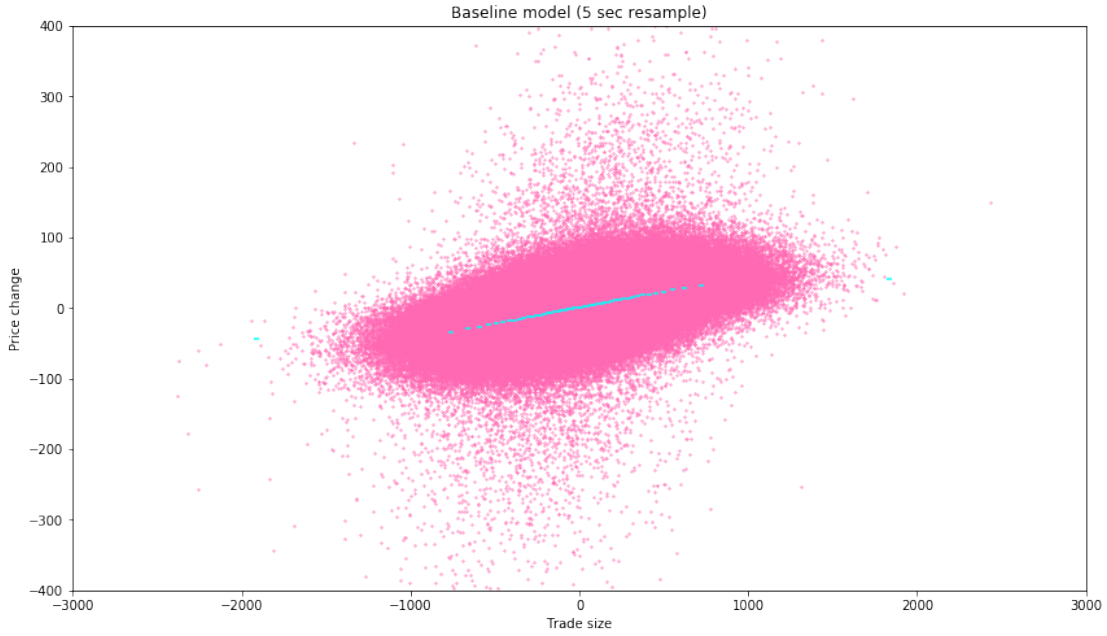

FIGURE 4.12: Baseline Market Impact (5 sec resample)

tuned only to passively observed stylised facts thus far, with no regard for the system's interactivity

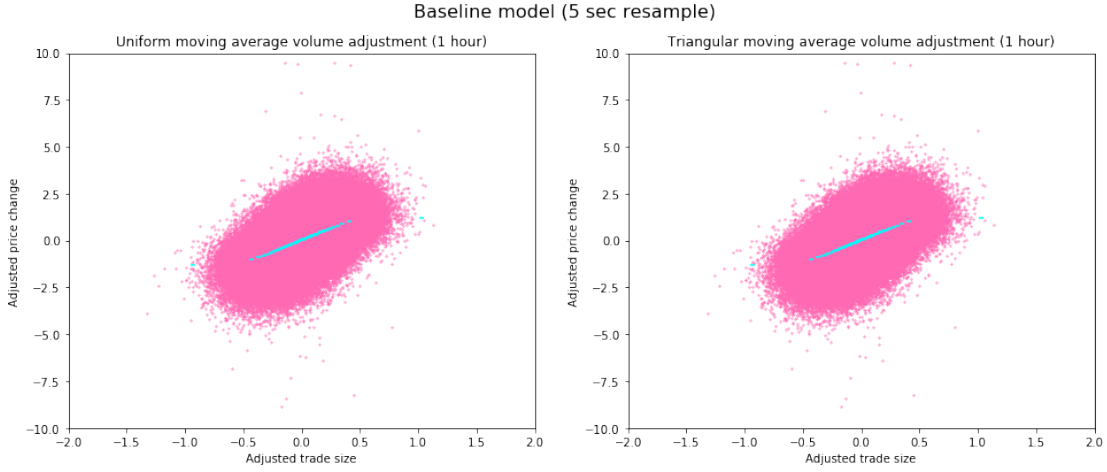

FIGURE 4.13: Baseline Market Impact with percentile bucket average impact (5 sec resample, volume & volatility adjusted)

This behaviour is far from the *square root law* that we are trying to replicate. Although it is likely possible to tweak the parameters of the participating agents in the baseline model to produce a more promising simulation result regarding the shape of market impact, this will come at a cost of additional assumptions and computation time. Therefore, we turn to different methodologies both existing and novel, with and without previous evidence of success, to replicate market impact.

4.4 Zero-Intelligence simulation

In order to try and replicate the *Square Root law* of market impact, we first return to the Zero-Intelligence (ZI) agents to observe the market impact these agents generate in a simple, homogeneous setting. By populating the ABIDES simulator with 1000 Zero-Intelligence agents with the same manual tuning applied in our optimal simulation from Section 3.4.8, we obtain the results seen in Figure 4.14. Note that the resampling period has been reduced from 5 seconds to 1 second to increase the number of data points collected from a smaller proof-of-concept simulation. We once again observe a linear relationship between trade size and returns prior to any adjustment, and it is clear that the ZI model by itself is insufficient to reproduce the *square root law* without modification. Therefore, we look at other methods outside of the basic ZI agents in an attempt to replicate the *square root law*.

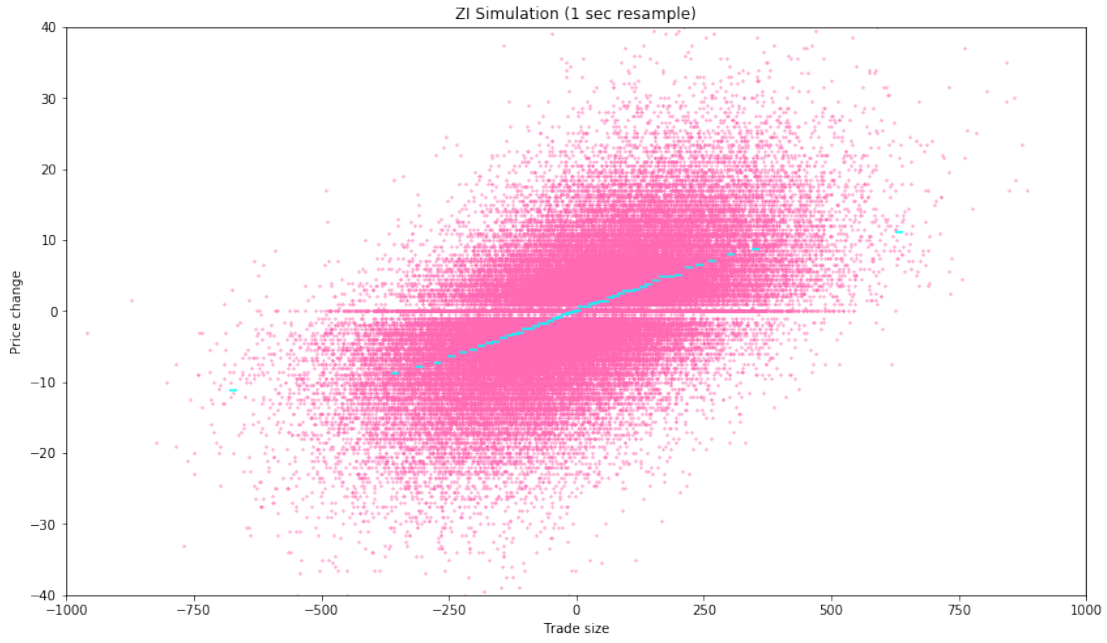

FIGURE 4.14: Zero-Intelligence BTC/USD Market Impact with percentile bucket average impact (1 sec resample)

4.5 Santa-Fe Model

In order to create a model that can replicate desired market impact, we turn to a proven method of modelling the market and its market impact. This model is called the *Santa-Fe Model* of the market (Farmer et al. (2005)), originally developed by researchers at the Santa Fe Institute. This model uses stochastic processes to populate the LOB following a zero-intelligence approach. Within this model, limit orders of fixed size arrive following a Poisson process across a predetermined price range. The arrival rate at each price

is independent of one another and independent of the number of outstanding orders at each level. At the same time, these limit orders are cancelled following a separate Poisson process, which is again independent in the same manner. Lastly, market orders arrive following yet another Poisson process that takes out limit orders around the mid-price. In doing so, the LOB is populated as a function of limit order arrival & cancellation rate, and market order arrival rate.

The *Santa-Fe model* can be replicated with small adjustments to the Zero-Intelligence agents, making it a logical follow-up from our previous modelling attempt using 1000 Zero-Intelligence agents in Section 4.4. The first step to replicate the *Santa-Fe model* using ZI agents is to divide these agents into limit order agents and market order agents. Although it is possible within the ABIDES platform to create a single type of Zero-Intelligence agent that is allocated a probability of behaving as a limit or a market order agent, this increases the run-time of the algorithm, without making any difference in the system behaviour. Therefore, we separate Zero-Intelligence agents into limit order agents (hereafter ZI-L) and market order (hereafter ZI-M) agents. While the agents here are similar to that of the Zero-Intelligence agent used in our simulation previously, the algorithm is slightly modified from Algorithm 1 to Algorithms 4 and 5. The ratio of ZI-L and ZI-M agents that we use for this simulation is 800:200, with the probability of cancellation $p_{cancel} = 0.2$.

Algorithm 4 Modified Zero-Intelligence Algorithm for Santa-Fe Model (Limit Order Agent)

$p_{cancel} := P(\text{Cancelling an existing limit order})$
Generate a random number $n \sim \mathcal{U}(0,1)$
if $n < p_{cancel}$ **then**
  Cancel oldest limit order if one exists
else
  Obtain private valuation, $V \sim \mathcal{U}(1,100)$
  Query best market ask, A'
  Query best market bid, B'
  Calculate mid price $M' = \frac{A'+B'}{2}$
  if $V < M'$ **then**
    Place limit order to buy at V
  else
    Place limit order to sell at V
  end if
end if

4.5.1 Initial Impact

As expected, the *Santa-Fe model* is able to reproduce the *Square root law*. The results in Figure 4.15 show a positive correlation between trade size and price change, without the need for any volume or volatility adjustment. When looking specifically at the bucketed

Algorithm 5 Modified Zero-Intelligence Algorithm for Santa-Fe Model (Market Order Agent)

1: Generate a random number $m \sim \mathcal{U}(0,1)$
2: **if** $m \leq \frac{1}{2}$ **then**
3: Place market order to buy
4: **else**
5: Place market order to sell
6: **end if**

mean price impact of trades (Figure 4.16), a non-linear relationship with diminishing impact towards larger trades can be observed, evidencing the existence of the exponent, δ, in Equation 4.1, and hence the reproducibility of the *Square root law*. Note that there is discreteness in the trade size dimension, as the trade size of the market-order agent is set to 1, which means the net traded value over any resampled period also ends up as a small discrete value.

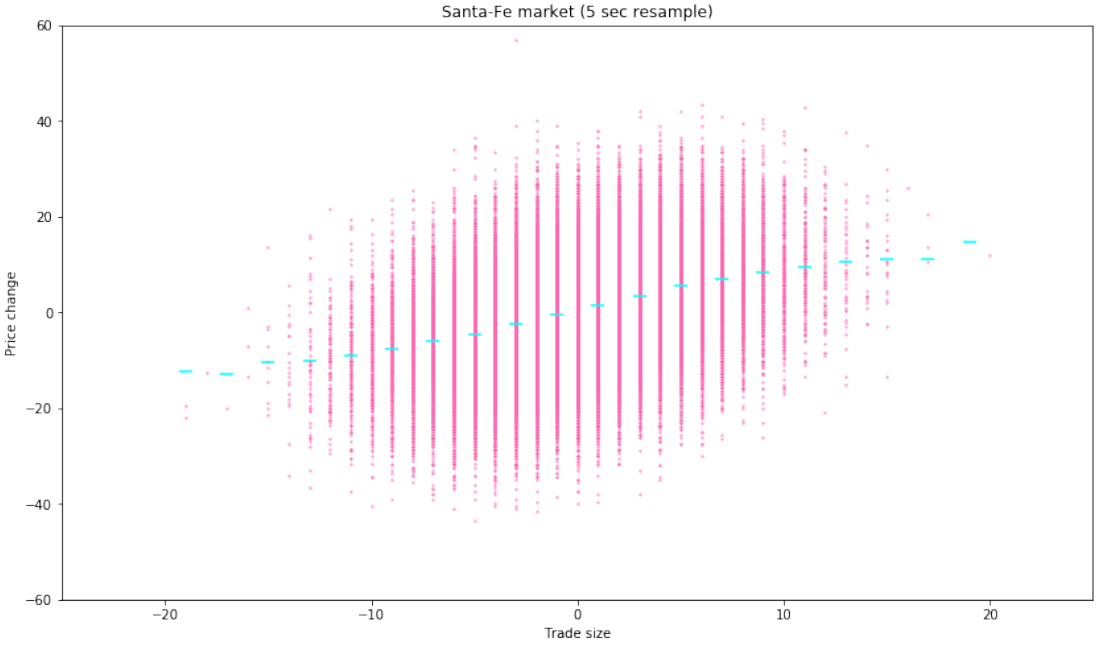

FIGURE 4.15: Santa-Fe Market Impact with percentile bucket average impact (5 sec resample)

4.5.2 Impact Reversion

On top of the *Santa-Fe model's* ability to reproduce the initial impact, it is also able to replicate the reversion of the initial impact attributed to the idea of the latent orderbook that we discussed in Section 4.2. This successful replication of reversion can be observed and argued in a similar fashion to the results shown in Figures 4.8, 4.9 and 4.10. By looking at the market impact of orders divided into previous order direction (Figure 4.17), as well as the regression coefficients of previous orders on the contemporaneous market

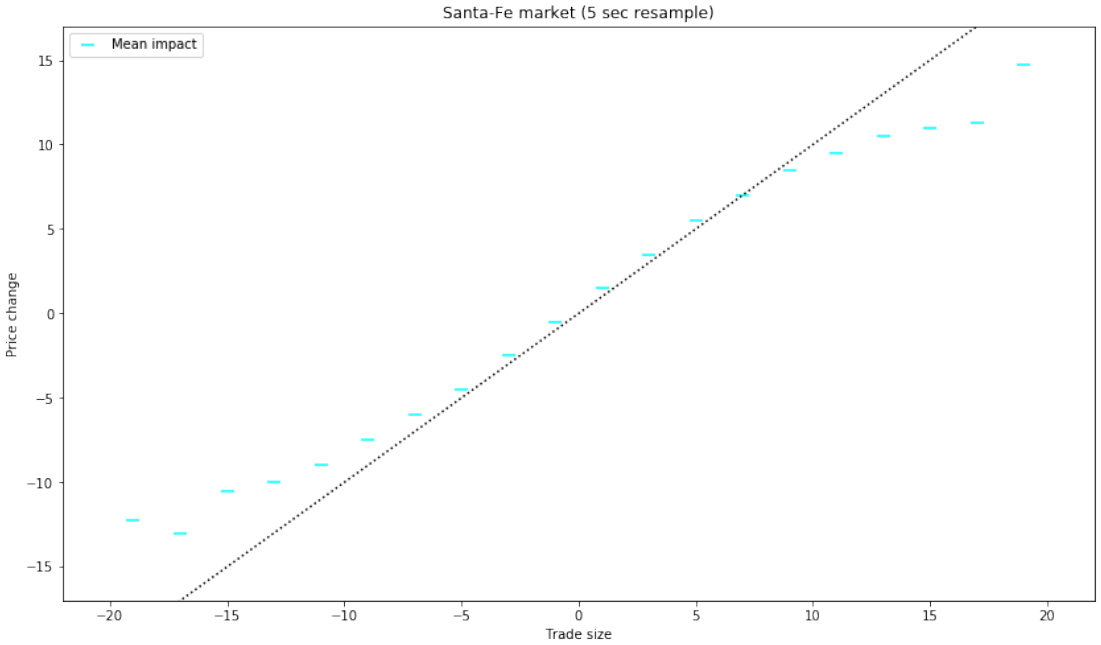

FIGURE 4.16: Zero-Intelligence BTC/USD Market Impact with percentile bucket average impact (1 sec resample)

impact (Figures 4.18 and 4.19), we can see the evidence of the market pushing against the initial impact over time.

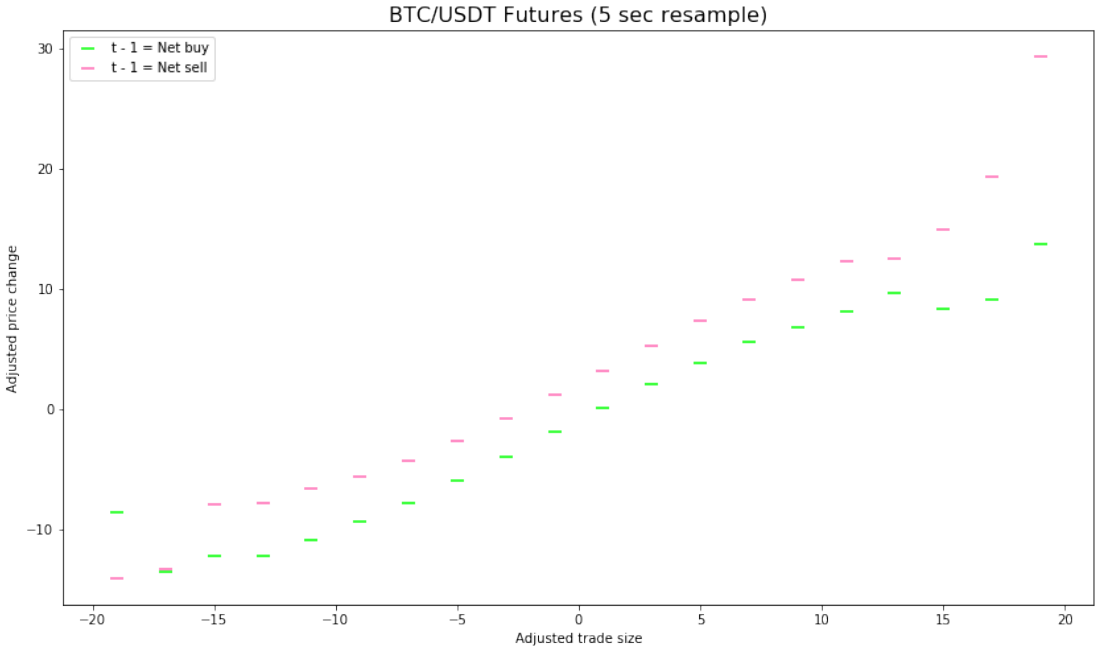

FIGURE 4.17: Santa-Fe Market Impact with percentile bucket average impact by previous order direction (5 sec resample)

The explanation for the existence of reversion within a *Santa-Fe model* is simple. As limit orders are taken out following a directional move in the market caused by a random walk (let us assume that in this case, buy orders consume the liquidity on the ask side

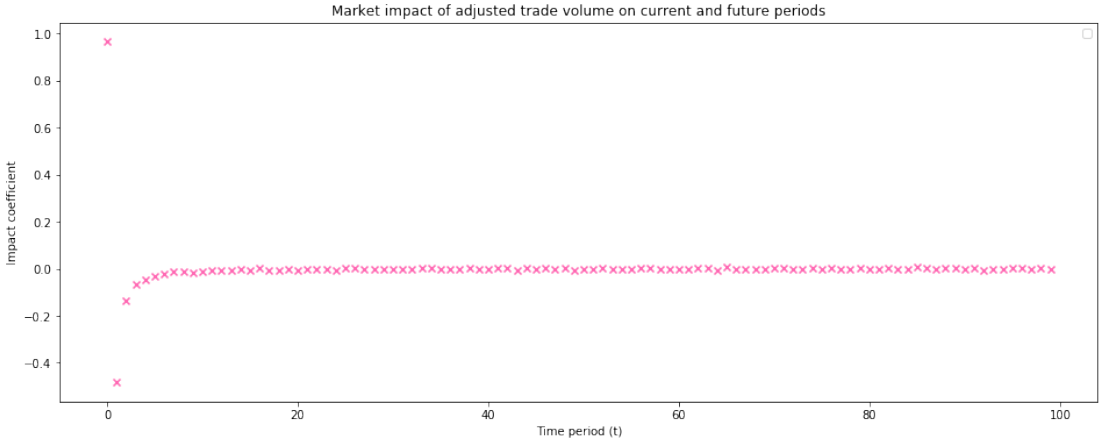

FIGURE 4.18: Santa-Fe market impact coefficient over time (5 sec resample)

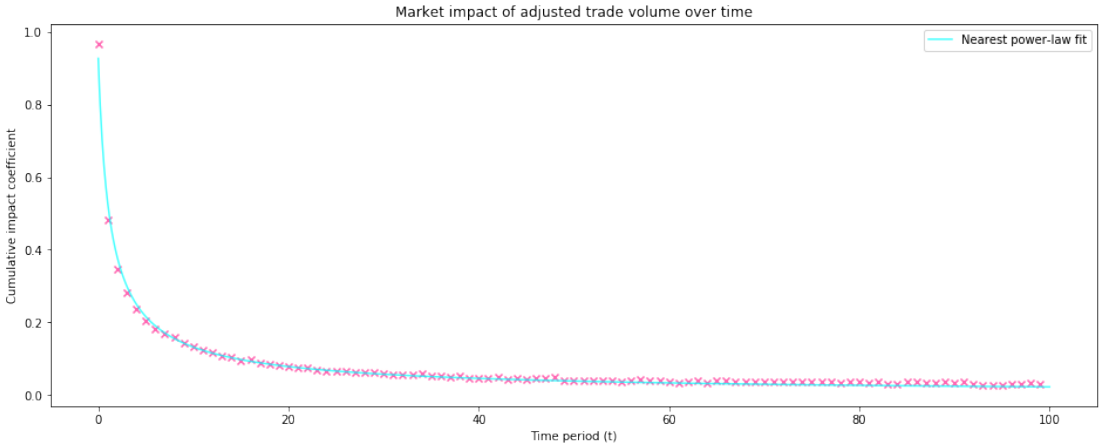

FIGURE 4.19: Santa-Fe market impact decay over time (5 sec resample)

of the LOB), an initial impact is caused as the best ask price increases while the best bid price remains constant. Following this move, the book continues to be filled and depleted on both the bid and the ask side of the LOB at the same rate. However, due to the increasing shape of the LOB, the starting state has more limit orders on the best ask price, in comparison to the best bid price. This means that the limit orders on the bid side are more likely to be depleted than on the ask side. Consequently, forthcoming market and limit orders will equilibrate the market further away from the best ask price (which has a higher outstanding volume than the best bid price), pushing the price down and reverting the upward impact caused by the initial transaction.

4.5.3 Meta Orders

As the *Santa-Fe model* is shown to replicate both initial impact and its decay effectively, we now take a look at its ability to replicate the idea of meta orders. As before, this is done by analysing the autocorrelation of order signs. Looking at the autocorrelation from the *Santa-Fe model* (Figure 4.20), we observe noise distributed around zero. This

is because the market orders arrive in this system in a random direction following a Poisson process, irrespective of the orderbook state. Therefore it is clear that whilst the *Santa-Fe model* is very much capable of generating the reversion effect of market impact, it cannot exhibit behaviour driven by meta orders.

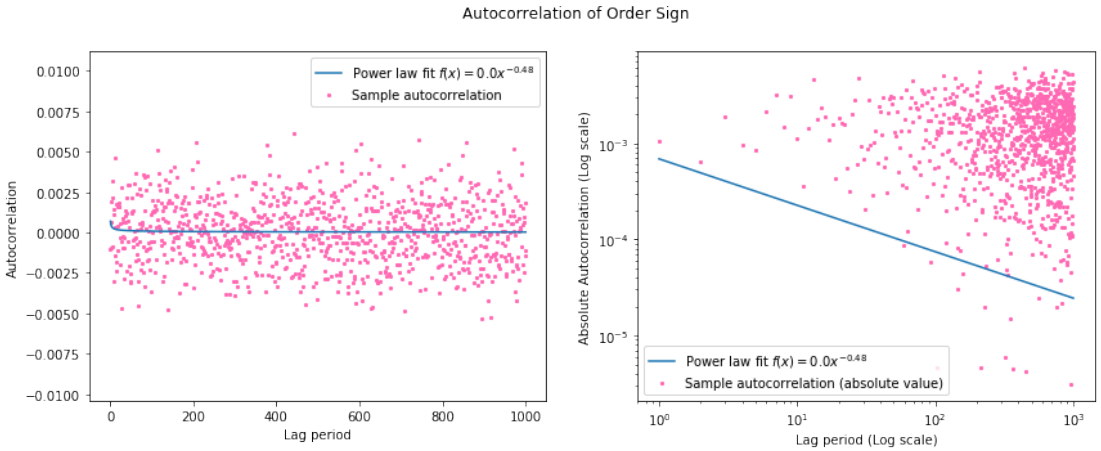

FIGURE 4.20: Autocorrelation of Order Sign (Santa-Fe Simulation)

4.5.4 Meta Order Augmentation

One of the ways to address the *Santa-Fe model's* inability to replicate order sign autocorrelation is to change the utility functions of market participants to trigger a desirable autocorrelative behaviour. However, as discussed previously in Section 3.4.3, while this is possible and desirable from an explainability perspective, adopting this method adds a large computational cost to the simulation, due to the increased agent complexity. Furthermore, specifying agent behaviour in a very particular manner makes the participant not very generalisable. Therefore, a different methodology that is computationally cheaper is considered.

The augmentation we apply to the *Santa-Fe model* is to alter the direction of market orders to be conditional on previous market orders, as suggested by Bouchaud (2018). This is achieved by generating the sign of the market orders following the DAR(p) process (Taranto et al. (2016)). This process generates a discrete autoregressive series, where autocorrelation decays over lags following a power-law, similar to that observed in the market (Figure 4.11). While the details of the process are discussed in detail by Taranto et al. (2016) in Section 2.4 of their paper, it is summarised again here to provide some intuition and to enhance reproducibility.

The DAR(p) process keeps track of the order signs of p previous orders at present time t. The sign at time t is seen as the "child" order of a previous "parent" order at time $t - l$, with l being a random variable following a discrete distribution λ_l, where $\sum_{l=1}^{\infty} \lambda_l = 1$,

subject to $\lambda_{l>p} \equiv 0$. Once the "parent" order is chosen following the λ_l distribution, the sign of the "child" order at time t, denoted as η_t, is determined by Equation 4.3.

$$\eta_t = \begin{cases} \eta_{t-l} & \text{with probability } \rho \\ -\eta_{t-l} & \text{with probability } 1-\rho \end{cases} \tag{4.3}$$

Taranto *et al.* also indicate that for the l^{th} order autocorrelation of order sign, η_t, to decay following a power law $l^{-\gamma}$ with exponent $\gamma < 1$, the following conditions must be met: $\lambda_l \sim l^{(\gamma-3)/2}$ and $\rho \to 1^-$. However, using this instruction blindly led to poor empirical results, where the decay in autocorrelation did not follow the decay observed in the live markets (Figure 4.11). The value of the exponent found in the market was approximately 0.47, but plugging in this value of γ and an arbitrarily large ρ value of 0.99 into the DAR(p) process led to insufficient decay in autocorrelation (Figure 4.21). This is likely due to the interaction between γ and ρ that occurs far away from the limit of $\rho \to 1^-$, as well as a disregard for the coefficient of the power law decay, which determines the first order autocorrelation of the system and, to a large extent, the magnitude of initial few terms.

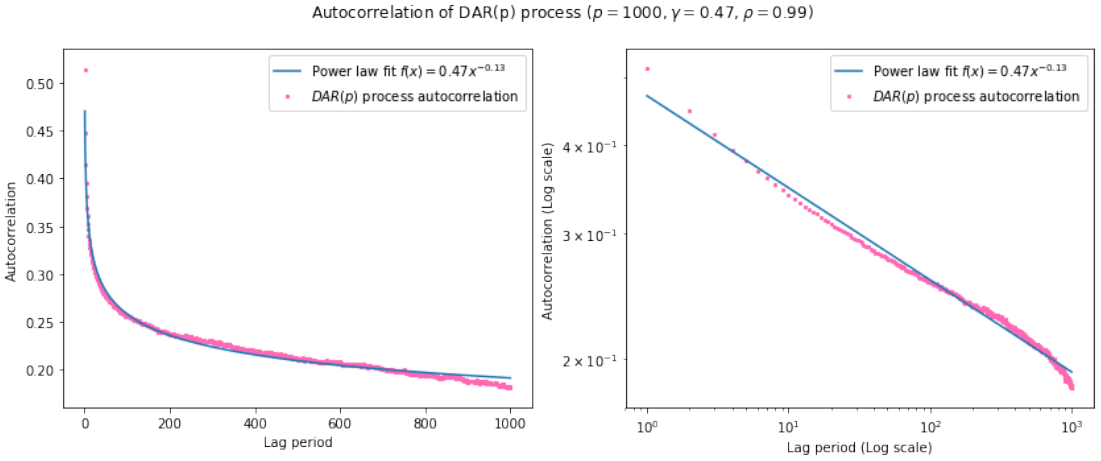

FIGURE 4.21: Autocorrelation of Order Sign (Simulated Data, March 2021)

Consequently, we use a brute force methodology to tune the parameters of the DAR(p) process. A grid search across the inputs of DAR(p) process, the γ and the ρ parameters, is carried out to reproduce the decay in autocorrelation seen in the BTC/USD exchange. The power-law decay, in the form of $f(x) = ax^{-k}$, that best fits the data collected from the BTC/USD exchange (Figure 4.11) has coefficient value $a \approx 0.76$ and exponent $k \approx 0.47$. By calculating the sum of squared losses (Equation 4.4) between these values and the parameters of the power-law fit on the decaying autoregression of the DAR(p) process, a heatmap can be generated to show the optimal region of the parameters (Figure 4.22). While a response surface of the two input variables to squared

loss can be created to estimate the parameters whilst reducing the impact of noisy simulation parameters, the optimal result from the grid search is taken without further augmentations. This is due to the trade-off between the bias caused by fitting a response surface to the data, and the higher variance caused by the randomness of simulation results. However, as the simulation is carried out over a relatively large sample size per parameter search (10^5) and the results appear reasonably smooth in Figure 4.22, more consideration was given towards reducing the bias from a fitted surface, over obtaining an "unlucky" simulation result. In addition, while a cross-validated grid search would reduce this "unluckiness" arising from the simulations, the current simulation running a 200 x 200 grid search in this parameter space of $-1.5 \leq \gamma < 0.5$ and $0.8 \leq \rho < 1$ already costs several hundred CPU hours. Consequently, the search was run only once in the interest of time and resources.

$$\ell = (a_{actual} - a_{DAR(p)})^2 + (k_{actual} - k_{DAR(p)})^2 \tag{4.4}$$

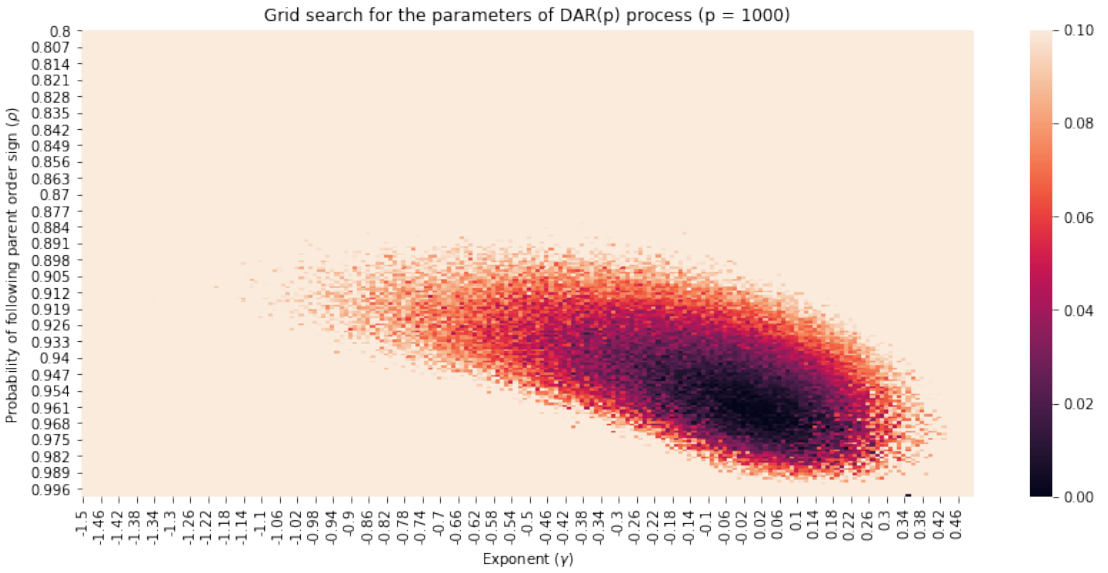

FIGURE 4.22: Grid search for the parameters of DAR(p) process

The DAR(p) process using the optimal parameters obtained from the grid search ($\gamma = 0.01$, $\rho = 0.965$) yields much better autocorrelation results than before (Figure 4.23). The resulting power law decay that best fits the data from the tuned DAR(p) process ($f(x) = 0.77x^{-0.45}$) is very similar to that from the market ($f(x) = 0.76x^{-0.47}$). At the same time, the order signs generated from the DAR(p) process remain equally likely to be a buy or a sell order as before. Therefore we now have a simulation that replicate the market impact and its decay following the Propagator model (Equation 4.2), whilst having the ability to represent the behaviour caused by meta orders.

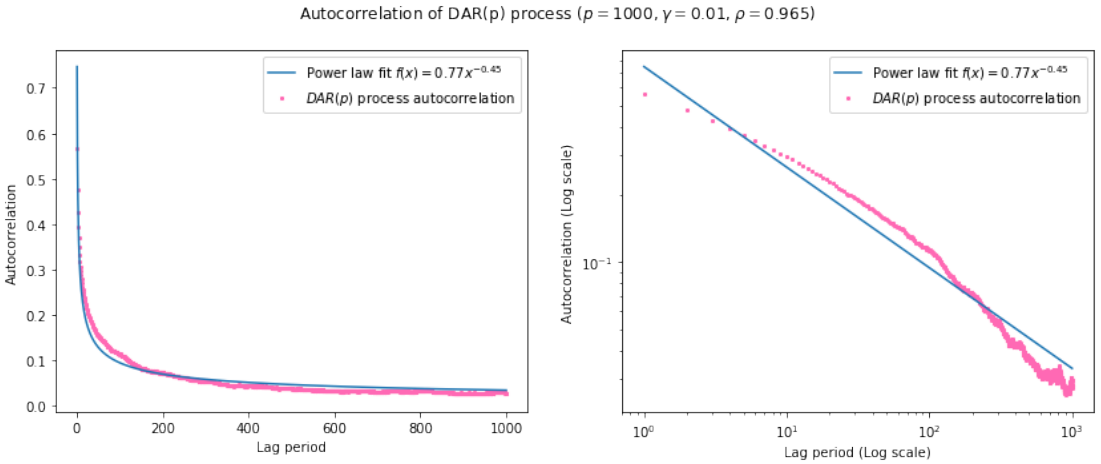

FIGURE 4.23: Autocorrelation of Order Sign (Simulated Data, March 2021)

4.5.5 Model evaluation

The *Santa-Fe model* is clearly able to reproduce the desired market response within the ABIDES platform. We see evidence of its desirable behaviour in reproducing the initial market impact of orders, the decay in market impact, as well as the autocorrelation of order signs, upon adding an adjustment to the model. Additionally, whilst we did not explicitly explore, we believe the magnitude and the functional form of the market impact and its reversion can also be tuned to be in line with the data observed in the BTC/USD Futures exchange, by tweaking the ratio of limit and market order agents.

However, despite the successful replication of these market microstructures, the model lacks a few key features for it to be used as a benchmark simulation. Firstly, the *Santa Fe model* lacks the ability to follow an underlying price series. Due to the fixed range of prices that the limit orders can be submitted to, alongside the random arrival of the market orders, whilst the system may mean revert or drift over time, it is inherently unable to follow an underlying price series. This is a problem, as participants being exposed to a directional change in underlying asset prices is an important feature of a market simulation and a key risk that participants must manage. In addition, the model suffers from a lack of explainability, as it assumes that orders submitted by participants of various different types can be modelled into two Poisson processes - the limit order process and the market order process. Lastly, augmentations such as the DAR(p) process forces the model to behave in a desirable manner. This forgoes the explainability of the system, as this behaviour is artificially induced by a stochastic process, as opposed to being an emergent behaviour as a consequence of the interactions between representative heterogeneous agents within the system.

Therefore, whilst the benefits of the *Santa Fe model* in reproducing desirable market microstructure are clear, due to its lack of explainability and the ability to follow an underlying price series, we look for a different way to model the market and incorporate

market impact into the system.

4.6 PRIME

In order to address the shortcomings of the *Santa Fe model*, we propose a new method of simulating market impact. It starts as an augmentation of the *Santa Fe model* that is approached from an agent-based perspective to incorporate both explainability and the ability to track an underlying price series. We call this model the **P**rice **R**everting **I**mpact **M**odel of a cryptocurrency **E**xchange, or PRIME.

The PRIME model is different from the *Santa Fe model* in a few ways. Firstly, limit order agents no longer "receive" a private valuation from a predetermined price range. Instead, they generate their private valuation by querying the prevailing mid-price of the simulation and adding a value drawn from a uniform distribution of predetermined width (Algorithm 6). Secondly, at the beginning of the simulation, the LOB is populated in a linearly increasing manner around a specified starting price. This allows the limit order agents to query the mid-price and populate the LOB from the beginning of the simulation, without running into a problem. Next, market order agents receive the true value of the underlying asset from the oracle, again with uniformly distributed perturbation. These agents then submit buy or sell orders to the market, based on their private valuation and the prevailing market price (Algorithm 7). Lastly, we embed the idea of "selective liquidity taking" in the market order agents within the PRIME model. This forces market order agents to place orders of varying sizes to the exchange, depending on the liquidity available at the top level. This idea was discussed in Bouchaud (2018), and we will explain it further during our implementation of it in Section 4.6.2 of our research.

These changes that make up the PRIME model lead to limit-order agents representing passive technical traders who act like background participants (or noise), and market-order agents representing the true fundamental agents, who only execute their strategies aggressively based on a valuation formed outside the market. In addition, the model also assumes that the limit order book is in a perfect state (with increasing volumes as we move away from the mid-price that plateaus) at the start point of the simulation.

When we start to make comparisons, PRIME behaves very similarly to the *Santa-Fe model* if the underlying value of the asset is fixed. In this case, limit orders arrive independently following a Poisson process on both sides of the prevailing mid-price, with an independent cancellation rate attributed to each order. market orders also arrive following a Poisson process, and if the prevailing market equilibrium is equal to the true underlying value, the model remains identical to the *Santa-Fe model*. However, if the model mid-price is not equal to the true price, the fundamental valuation these agents

Algorithm 6 Modified Zero-Intelligence Algorithm for PRIME Model (Limit Order Agent)

$p_{cancel} := P(\text{Cancelling an existing limit order})$
Generate a random number $n \sim \mathcal{U}(0, 1)$
if $n < p_{cancel}$ **then**
  Cancel oldest limit order if one exists
else
  Obtain perturbation, $\epsilon \sim \mathcal{U}(-50, 50)$
  Query best market ask, A'
  Query best market bid, B'
  Calculate mid price $M' = \frac{A'+B'}{2}$
  Calculate private valuation $V = M' + \epsilon$
  if $V < M'$ **then**
    Place limit order to buy at V
  else if $V > M'$ **then**
    Place limit order to sell at V
  else
    Do not place limit order
  end if
end if

Algorithm 7 Modified Zero-Intelligence Algorithm for PRIME Model (Market Order Agent)

$W :=$ Size of the observation error
Receive true price, P, from the Oracle
Obtain perturbation, $\epsilon \sim \mathcal{U}(-W/2, W/2)$
Calculate private valuation, $V = P + \epsilon$
Query best market ask, A'
Query best market bid, B'
Calculate mid price, $M' = \frac{A'+B'}{2}$
if $V > M'$ **then**
  Place market order to buy
else if $V < M'$ **then**
  Place market order to sell
else
  Do not place market order
end if

receive will be skewed away from the simulation's mid-price. This results in the market orders arriving in a skewed manner (asymmetric buy/sell order arrival rate) to push the model equilibrium back towards the true price. This adds a mean reverting non-drift feature to the PRIME model, which does not explicitly exist in our implementation of the *Santa-Fe model*. Conversely, if the true price of the underlying asset is not fixed at the initial price and instead fluctuates following a predetermined path (e.g., historical price series), then PRIME offers a simulation that follows the underlying price series, whilst largely maintaining the market mechanics of the *Santa-Fe model*. In addition, due to the ability of PRIME to follow and revert back to the "true" market price, it is able to interact

with external agents that may initially push the system out of equilibrium. These factors make the PRIME model much more useful when replicating the real-world market in an explainable manner, as we are able to add various types of agents to our simulation without the worry of the system deviating too far from the underlying price series.

In the remainder of this section, we provide evidence of the desirable behaviour exhibited by the PRIME model that we have claimed thus far.

4.6.1 Price Tracking

The first evidence we present is the PRIME model's ability to track and revert to an underlying price series. As shown in figure 4.24, the PRIME model is able to track the underlying price series quite closely, even in the presence of observation error. This is due to the market order agents (on average) exerting pressure on the system towards the underlying price series. As the observation error of these agents is reduced, we can see the system equilibrating closer to the underlying series, as the direction of the pressure applied by the market order agents will be more unified towards the true price, as opposed to being more of a stochastic pressure that is on average towards the true price. This behaviour can be seen in Figure 4.24, where the mid-price of the PRIME simulation tracks the actual price series fairly closely over 10 minutes, across all levels of observation error.

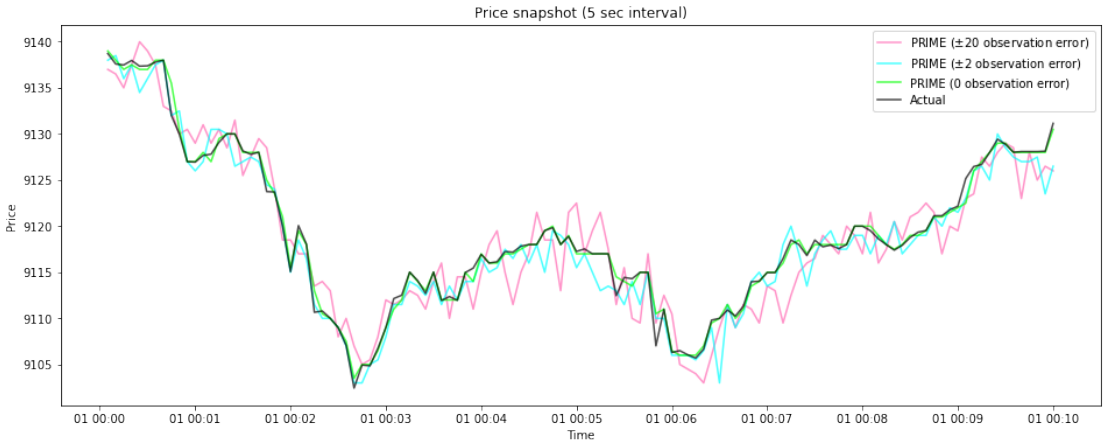

FIGURE 4.24: Price paths (5 second snapshot, varying observation error, July 2020)

When the simulation's ability to track the price is viewed from a price deviation perspective (Figure 4.25), the 3 standard deviation error margin (in shaded areas) around the mean of each simulation with different observation error reveals what we expect. The lower the margin, the more likely is the difference between the actual and simulated mid-price to be smaller.

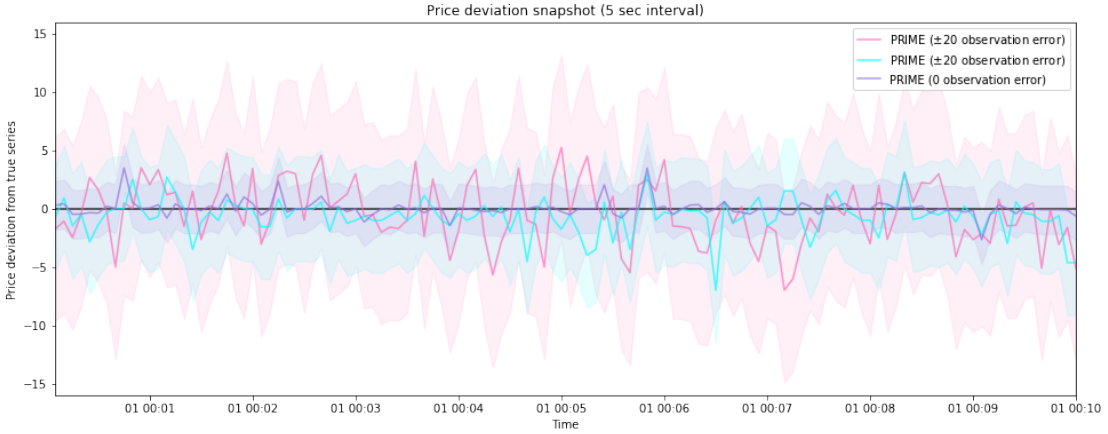

FIGURE 4.25: Price deviation (5 second snapshot, varying observation error, July 2020)

Another interesting observation we made from the price tracking behaviour of the PRIME model is the relationship between rate of convergence and the ratio of the limit and market order agents. A greater ratio of market order agents leads to quicker equilibration of the market towards the true underlying price of the market. However, this faster equilibration occurs at a much lower ratio of market order agents than we anticipated. In a simulation with 1000 limit order agents, the benefit of having 20 market order agents and 50 market order agents is not pronounced (Figure 4.26 and Figure 4.27), especially as the simulation moves away from the start time, where the LOB is filled with linearly increasing limit orders from the mid-price.

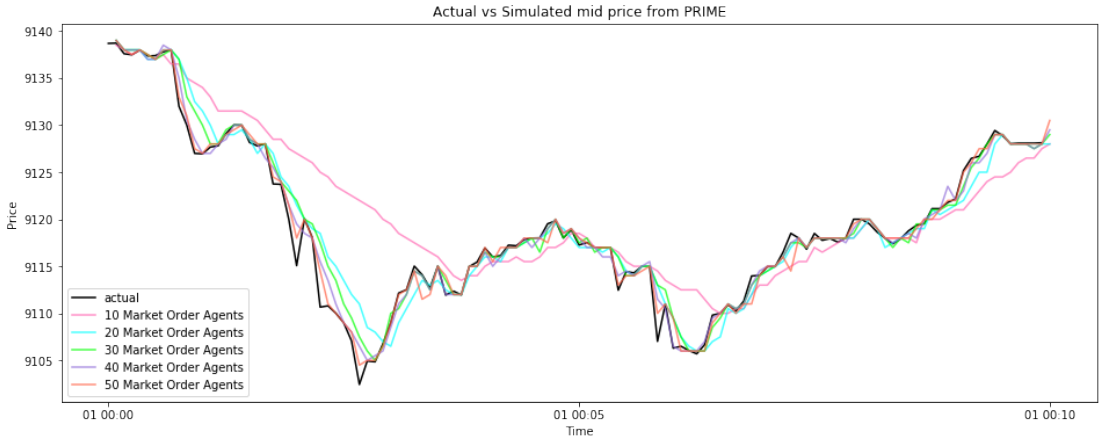

FIGURE 4.26: Price deviation (5-second snapshot, 1000 limit order agents, July 2020)

To allow for some perturbation of the simulation's price series, whilst maintaining its ability to follow the underlying true price series, we choose to adopt an observation error margin of ± 20, and place 30 Zero-Intelligence market order agents into the system (Green line, Figure 4.26 and Figure 4.27). We recognise the arbitrary nature of this decision, but we believe this configuration allows the simulation to be distorted upon large shocks, thus retaining the ability to effectively replicate the real markets, whilst maintaining the ability to revert back to the true price.

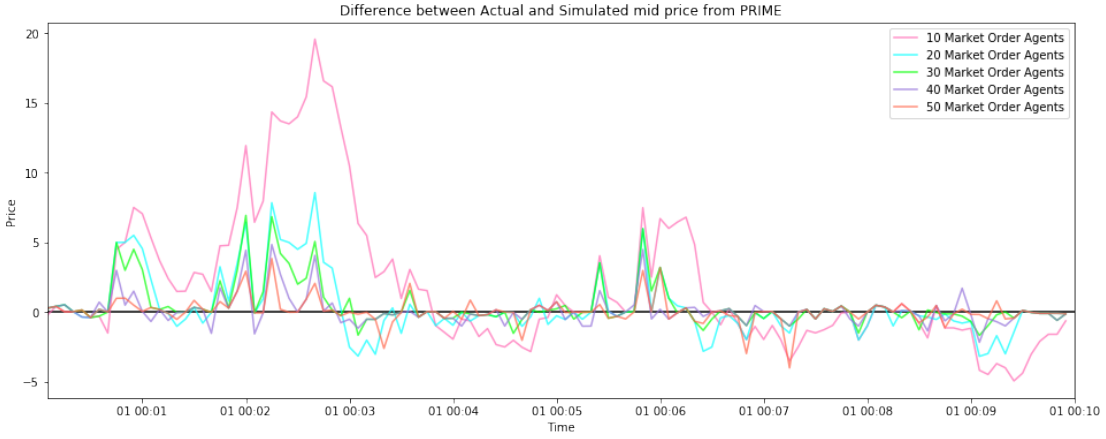

FIGURE 4.27: Price deviation (5 second snapshot, 1000 limit order agents, July 2020)

4.6.2 Selective Liquidity Taking

The idea of selective liquidity taking was mentioned in the previous section and is embedded in the zero-intelligence market order agents that form the PRIME model. This refers to market participants adjusting the size of their market orders depending on the available liquidity at the top level of the LOB. This is a very intuitive behaviour, since removing some fraction of the liquidity at the top level results in the transaction being carried out at the same price, whilst not having a direct impact on the mid-price. This suggests that participants are likely to take more liquidity from the market if more volume is available to trade at the top level. To reproduce this behaviour, we use the same simple volume function introduced by Bouchaud (2018), where the size of the market order is an increasing function of the prevailing volume at the top level (Equation 4.5). In our simulation, we use the parameter $\nu_0 = 1$ to reflect the smallest order size accepted by the ABIDES simulator. One point to note as we incorporate selective liquidity taking is the introduction of a non-linear relationship between the number of market order agents and market order volume, and thus the prevailing liquidity. Previously, doubling the number of market order agents increased the number of market orders by twofold. However, with selective liquidity taking, increasing the number of agents leads to faster depletion of top-level liquidity, which in turn puts negative pressure on the liquidity consumed per order. Therefore, the behaviour of the system will be harder to estimate without empirical analyses of the simulation results.

$$V_{market\_order} = \nu_0^{1-\Psi} V_{top\_level}^{\Psi} \tag{4.5}$$

We analyse the behaviour of the *PRIME* model after incorporating selective liquidity taking to the market order agents. One obvious change following the move away from market orders of unit size to a function of the volume at the top level is the model's ability to track the underlying price series. As shown in Figure 4.28 and Figure 4.29, the

greater the volume taken away from the top level, the faster the simulation converges to the underlying price series. This is unsurprising, as selective liquidity taking increases the volume of market orders as the value of Ψ increases. In addition to this behaviour, we observe that at a value around $\Psi = 0.5$, the model does a good job converging to the underlying price series irrespective of its rate of change.

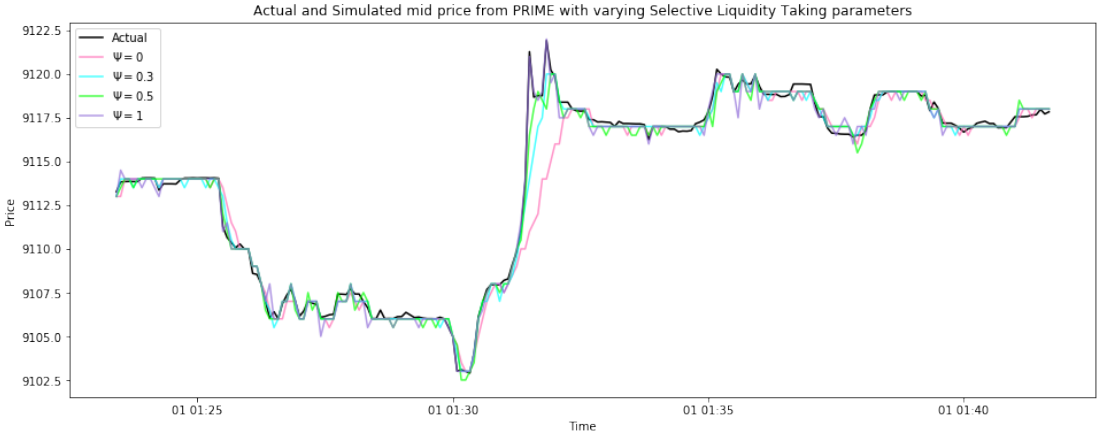

FIGURE 4.28: Autocorrelation of Order Sign (Simulated Data, March 2021)

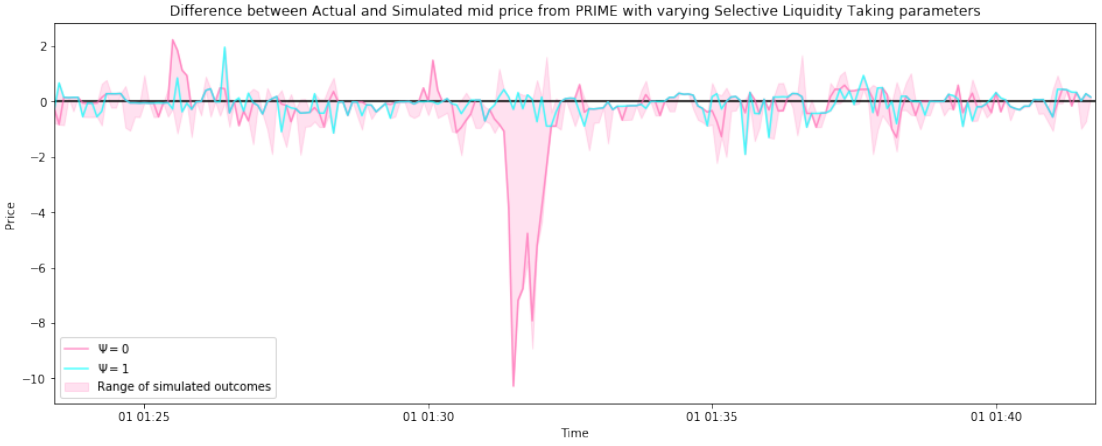

FIGURE 4.29: Autocorrelation of Order Sign (Simulated Data, March 2021)

Another interesting observation we made after augmenting the model was the change in order sign autocorrelation after implementing selective liquidity taking (Figure 4.30). At the constant ratio of zero intelligence agents of 1000 limit order agents and 30 market order agents adopting selective liquidity taking, varying the exponent parameter which determines the proportion of top-level liquidity that is taken, Ψ, had little influence on the rate of decay of power law. However, it did have a material impact on the magnitude of the initial correlation, most likely due to the increased number of market orders in the same direction required to clear the top level to move the simulation price back in line with the underlying price series.

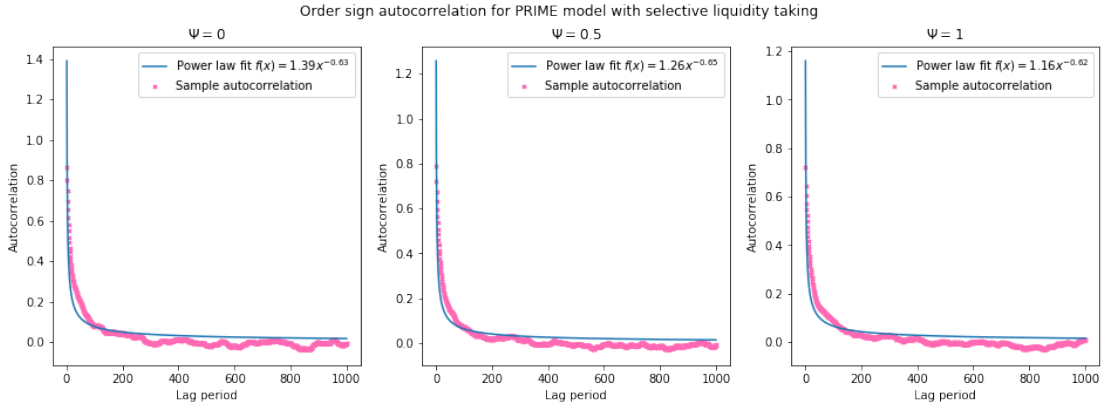

FIGURE 4.30: Autocorrelation of Order Sign (Simulated Data, July 2020)

4.6.3 Initial Impact

Here, we analyse the initial impact of trades observed in the PRIME simulation using the same methodology as in Section 4.2. Before including any of the adjustments using the *square root law* of market impact (Equation 4.1), we observe a wide range of impact outcomes for periods of smaller trade volume, and a smaller range for trades with larger trade volume (Figure 4.31). This is somewhat in line with our observations from the BTC/USD Futures market (Figure 4.2).

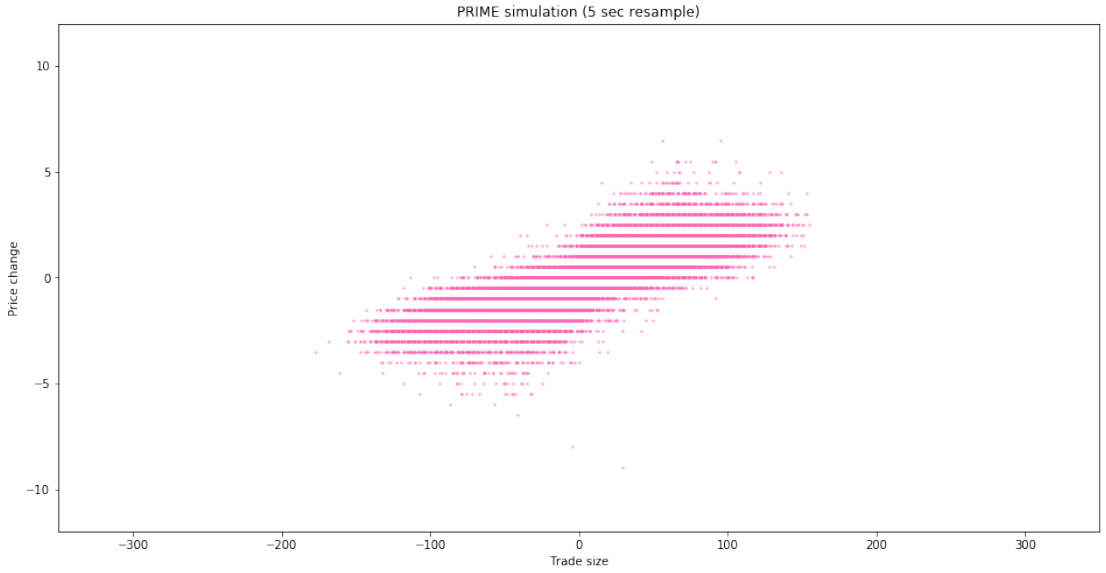

FIGURE 4.31: PRIME Market Impact (5 sec resample)

There are two major observable differences between our simulation and the BTC/USD Futures market. Firstly, we observe a lack of low-volume high-impact data points, which in the real market often occur during periods of low liquidity, such as time outside of the major market hours. This difference is unsurprising since our simulation does not have periods of low liquidity built into the system and instead maintains a constant level of market liquidity throughout the simulated timeframe. Secondly, we notice that the

scatter plot is centred around a positive net trade size, as opposed to a zero net trade size. This can be observed more clearly in a histogram of net trade sizes (Figure 4.32), where the simulation shows a clear positive bias for net trade volume. Although we were initially concerned about the system's bias towards buying the asset, it turned out to be just a side-effect of the underlying price series that our model was following being on an upward trend (Figure 4.33, left image). Upon running the simulation over a time period with falling prices, we see a negative bias in net trade appearing, suggesting that the model in itself is not biased towards either buy or sell orders.

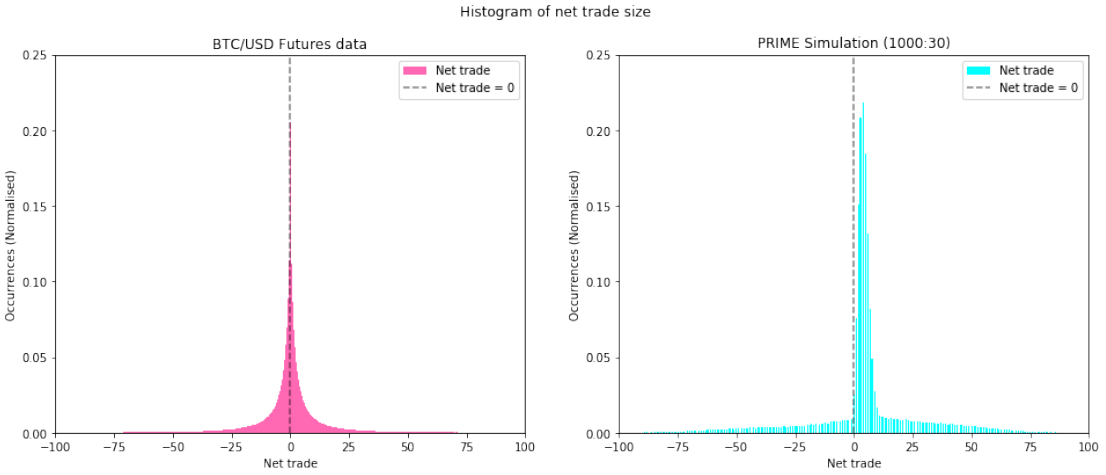

FIGURE 4.32: PRIME net order volume histogram (5 sec resample)

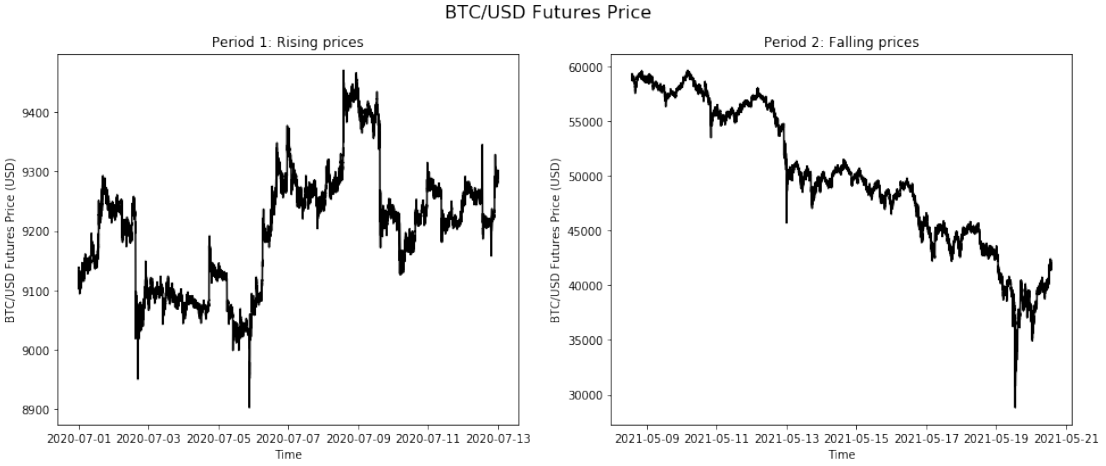

FIGURE 4.33: PRIME net order volume histogram (5 sec resample)

Upon adding both volume and volatility adjustments, the results of the PRIME simulation become very promising (Figure 4.34). Although the resulting scatter plot is sparse near the region of small adjusted volume and large adjusted impact (this is likely due to the aforementioned lack of low volatility periods), the underlying relationship appears linear, and more homoskedastic than before. Looking closer at the data using bucketed means as per Figure 4.35, we observe a non-linear relationship between trade size and price change that diminishes as trade size becomes larger. This evidences the ability of

the PRIME model to reproduce the diminishing market impact of orders, and therefore the replication of the desired *square root law* of the initial impact.

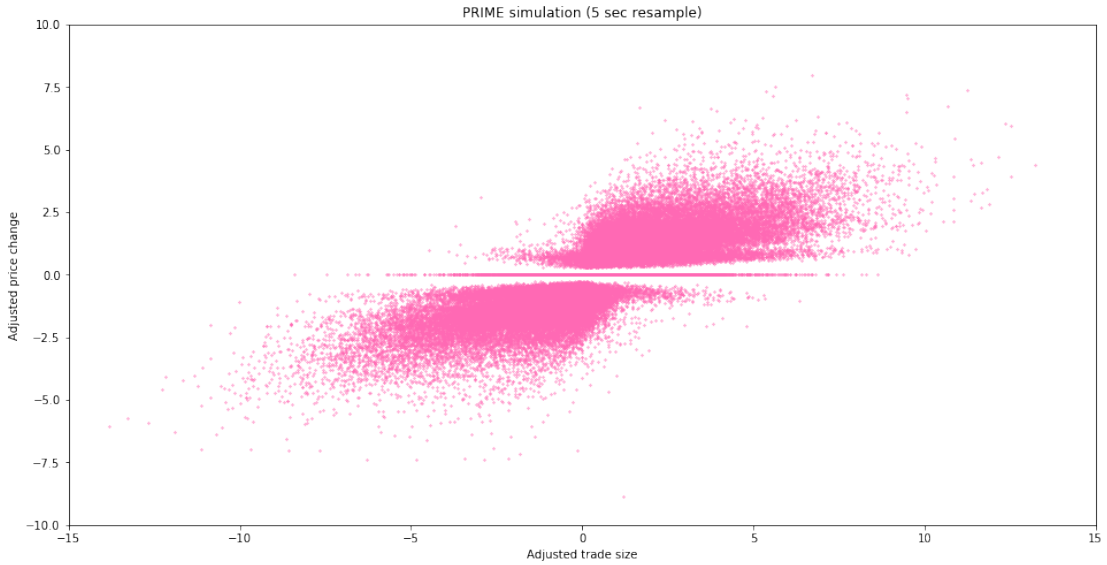

FIGURE 4.34: PRIME Market Impact (5 sec resample, adjusted)

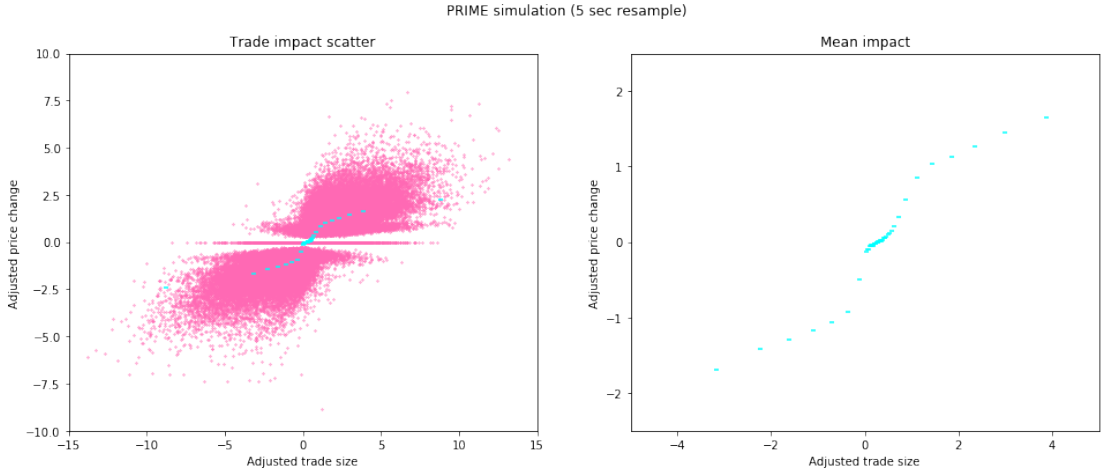

FIGURE 4.35: PRIME Market Impact (5 sec resample, adjusted)

4.6.4 Impact Reversion

We next look at the model's ability to replicate the reversion in market impact. As before, we approach reversion from two perspectives: the conditionality of previous order direction on current impact and the regression coefficients of previous orders on current market impact. First, we show evidence of reversion based on the conditionality of the previous order direction on the current market impact within the *PRIME* model (Figure 4.36). As before, when the data is divided into orders following a previous period of buy orders and sell orders, the previous buy orders exert negative pressure on the current market impact and vice versa. Although the evidence is not clear around the

0.1 adjusted trade size mark where trades are concentrated, the data from larger trade sizes suggests that the model is able to reproduce reversion for larger trade sizes.

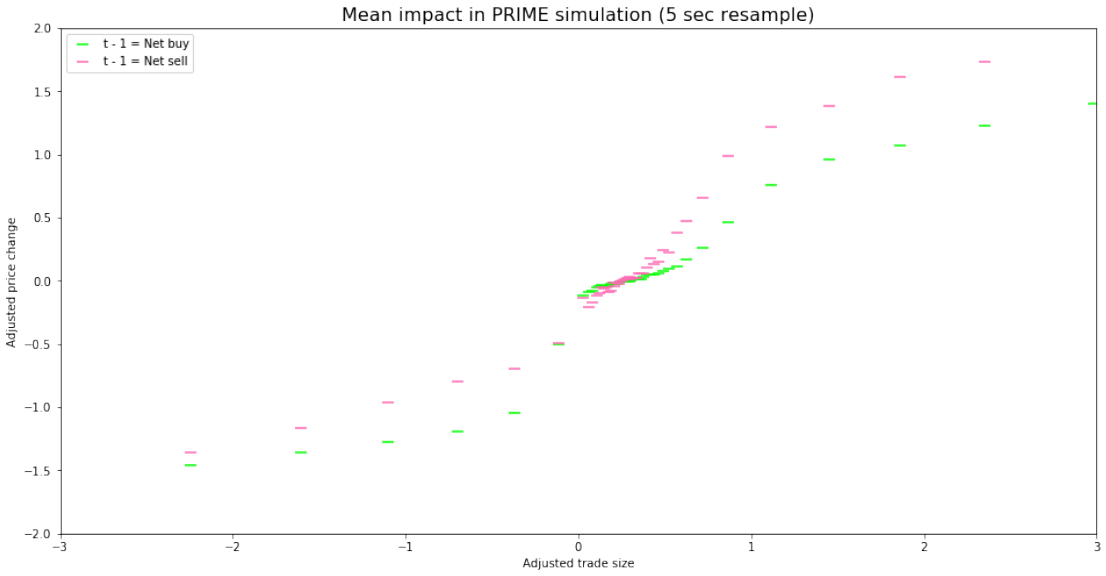

FIGURE 4.36: PRIME market impact coefficient over time (5 sec resample)

The second method of measuring the reversion in market impact using lagged regression also show support of reversion within the PRIME framework. The regression coefficients of trades on future impacts (Figure 4.37) behave generally in line with expectation, with a large initial impact, followed by periods of negative impact, which decays over time. Looking at the cumulative impact of trades over time (Figure 4.38), it initially decays in a manner that could be modelled using the power law, but the decay becomes linear after the first few lags. Although the shape of the decay is not exactly in line with the power-law decay observed in the real world (Figure 4.10), the results are promising. However, there is one data point that is difficult to explain. The market impact of the trade in the second time period should be a large negative. However, the *PRIME* model yields a small negative market impact. This goes against real-world data and market intuition, as there is no reason for the reversion to skip a time period.

We believe that this is the result of a high correlation of the net trade direction between two consecutive periods. In other words, the 5-second window that we specify as a single "period" may not be suitable to break the market into discrete independent events. To test this belief, we resampled the trade data into larger 10-second buckets and ran the same regression analysis to look for evidence of reversion. We present the findings in Figure 4.39 and Figure 4.40. Looking at the results, the second-period impact now shows the largest negative value, indicative of a strong reversion, and the remaining decay follows a similar shape as before, albeit with a little more noise. This suggests that the reversion in the *PRIME* model does behave in an expected fashion, and it is the discreteness of our analysis, not the model, that is yielding a counter-intuitive behaviour.

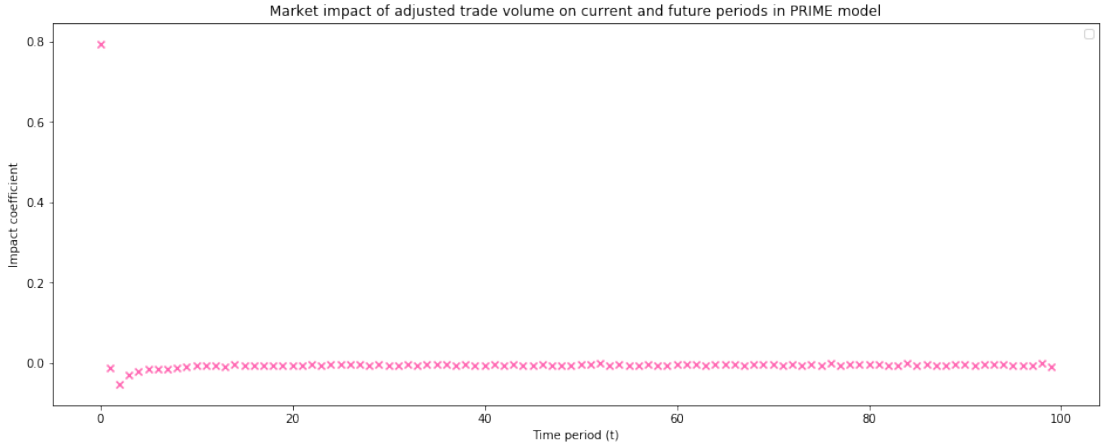

FIGURE 4.37: PRIME market impact coefficient over time (5 sec resample)

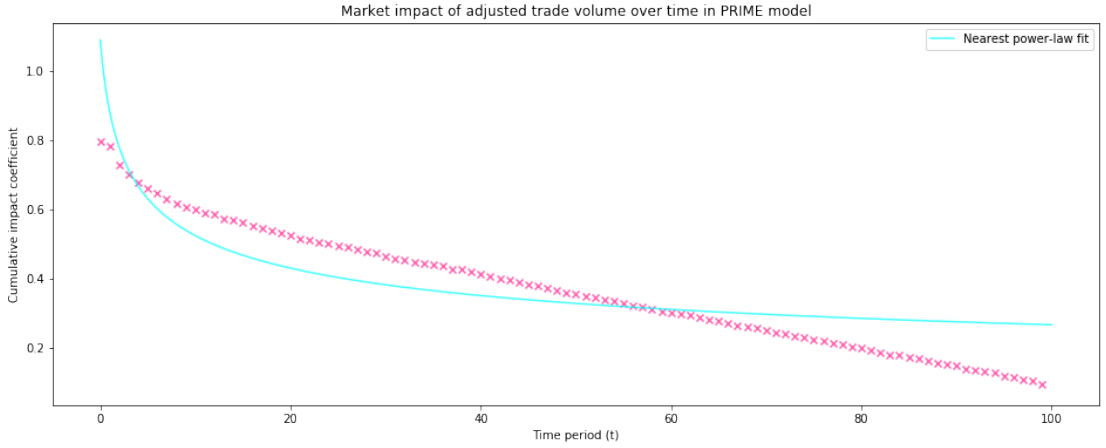

FIGURE 4.38: PRIME market impact coefficient over time (5 sec resample)

Therefore, despite the slight discrepancy we observed previously, we accept the above results as good evidence of reversion in market impact.

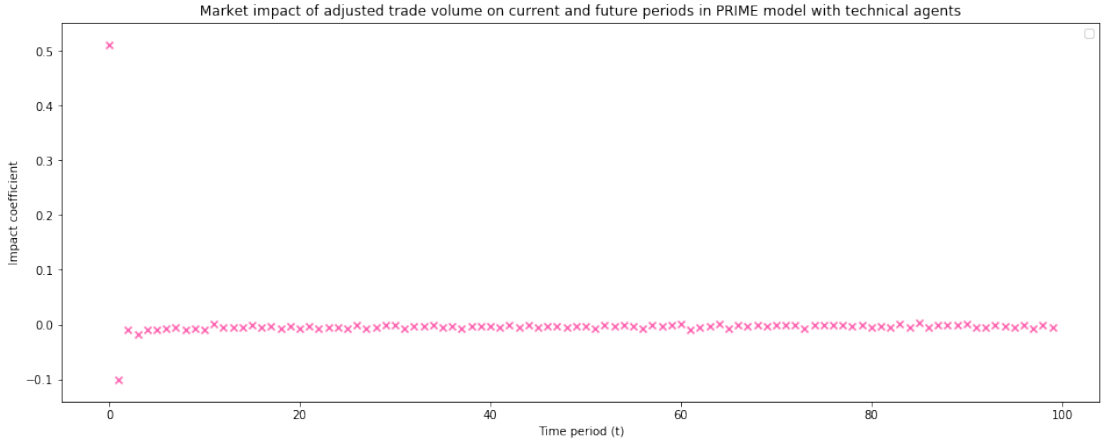

FIGURE 4.39: PRIME market impact coefficient over time (10 sec resample)

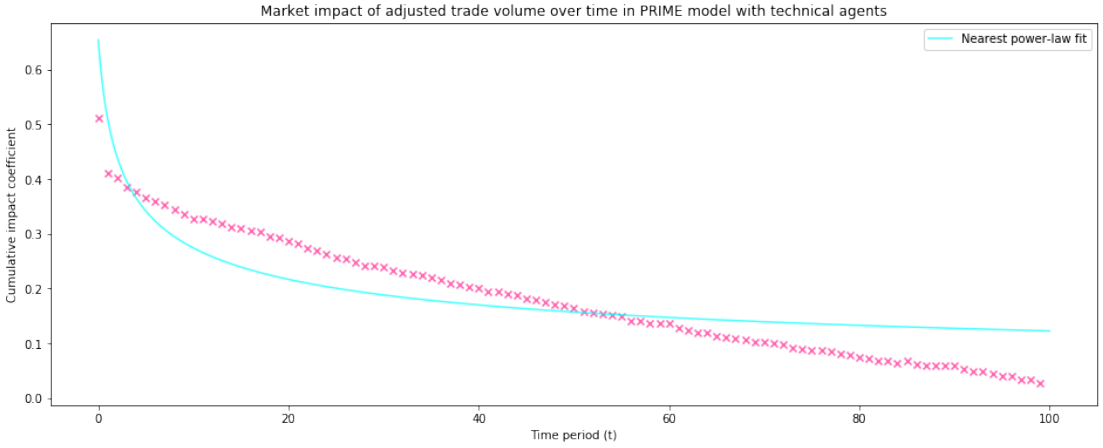

FIGURE 4.40: PRIME market impact coefficient over time (10 sec resample)

4.6.5 Meta Orders

As the final step in our assessment of the *PRIME* model, we examine the autocorrelation of order signs from the trade data. Here, we see a promising result, as shown in Figure 4.41. Like the real-world exchange, we observe a decay that roughly follows the power law. This is especially encouraging, as the *PRIME* model does not embed unexplainable stochastic processes into the participating agents to generate this result. We believe that this autocorrelation occurs from the mean-reverting behaviour of the system, where market orders arrive in the same direction, following a deviation of the simulation's mid-price from the fundamental agents' underlying price series. While this does not correspond exactly to the idea of meta orders, this is an improvement over the empirically fitted stochastic process adopted by the augmented *Santa-Fe model*.

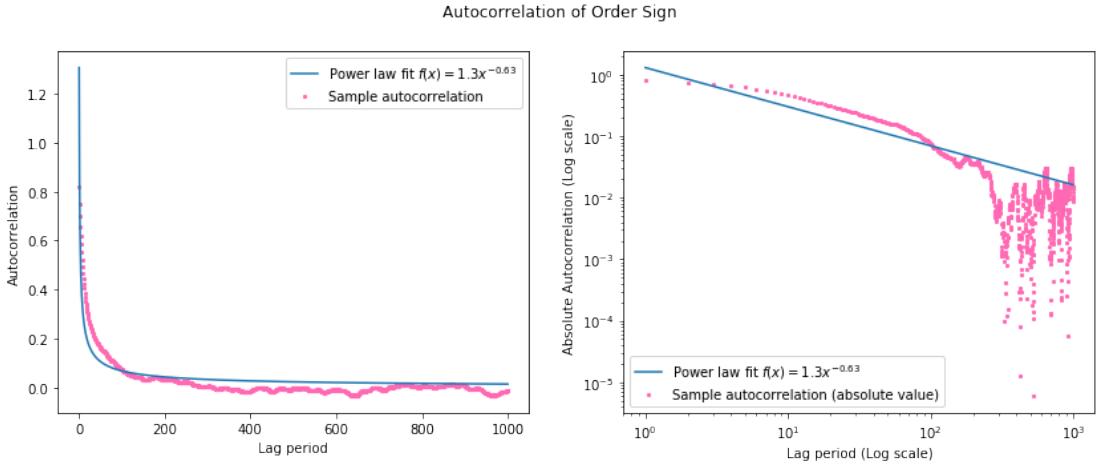

FIGURE 4.41: PRIME autocorrelation of Order Sign (Simulated Data, March 2021)

4.6.6 Adding technical agents

As discussed previously, the *PRIME* model has the ability to revert back to an underlying price series following an exogenous shock to the system. Although a shock will cause an immediate deviation from the price series, the magnitude of this deviation will decay over time, as market orders push the system back towards the price series and limit orders repopulate the LOB to the equilibrium level. In turn, this allows the users to add various other agent types to interact with the system, whilst maintaining the general price trajectory of the simulation. This is key to the PRIME model's enhanced explainability. By adding agents that can represent different categories of market participants that we used previously in Chapter 3, the model can produce a more desirable behaviour. As the PRIME model currently is driven by what we define as fundamental agents and noise agents, we now add technical agents, "Momentum" and "Mean-Reversion" to the simulation to represent market participants implementing endogenous technical trading strategies. We report and analyse the emergent behaviour of the simulation upon including these technical agents in this subsection. Note that the current iteration of technical agents submits only market orders to the exchange, reducing the equilibrium level of liquidity in the LOB, whilst increasing the market volatility.

The agent configuration we use is 1000 zero-intelligence limit order agents, 30 Zero-Intelligence market order agents, 10 momentum agents, and 10 mean-reversion agents. In essence, we have taken the 1000/50 configuration that we used in the previous sections as the baseline and substituted the zero-intelligence agents that place market orders with technical agents to analyse their impact on the simulation.

4.6.6.1 Initial Impact

The first thing we notice when adding technical agents is the improved shape of the adjusted market impact plot. The heteroskedasticity of the relationship between the order size and the price change has disappeared largely, as evidenced by the rectangular shape of the scatter plot in Figure 4.42. We initially suspected that the change in shape was due to the difference in the total size of the market orders that are submitted to the system. This hypothesis was investigated by running the *PRIME* model with varying levels of zero intelligence market order agents (Algorithm 7), rather than adding technical agents. The results (Figure 4.43) show some visual improvements as the number of market order agents increases, but their degree of improvement is not nearly as impactful as the introduction of technical agents seen in Figure 4.42. This indicates that the inclusion of technical agents influences the impact through a channel other than the order volume.

We believe that this difference occurs because of the changing nature of time periods in which net traded volume is near zero. With the inclusion of mean-reversion agents,

FIGURE 4.42: PRIME Market Impact (5 sec resample, adjusted)

FIGURE 4.43: PRIME Market Impact Comparison (5 sec resample, adjusted)

periods of zero net traded volume are likely to have higher total traded volume (sum of all buy and sell orders), as these agents are likely to cancel out directional moves. However, submitting 100 buy orders followed by 100 sell orders leads to different price results than submitting 10 orders on each side, due to various dynamics of the LOB. In fact, the former is more likely to cause a larger price impact than the latter, since the impact on the two sides of the LOB is much more likely to be asymmetric. Upon investigation, we find evidence of an increase in total trade volume for the middle 5% of net traded volumes (which roughly corresponds to net trade volume around zero). This is shown in Figure 4.44, where the total trade volume in the simulation, including technical agents (right graph), has a higher mean total trade size. Likewise, the price change for the same middle 5% of the net traded volume has a higher standard deviation (Figure 4.45), indicating a larger range of price impacts around zero net volume. This provides an explanation behind the more desirable simulation results discussed above.

In addition to the improved simulation results observed on the scatter plot, the addition of technical agents also appears to improve the shape of the *square root law* discussed in

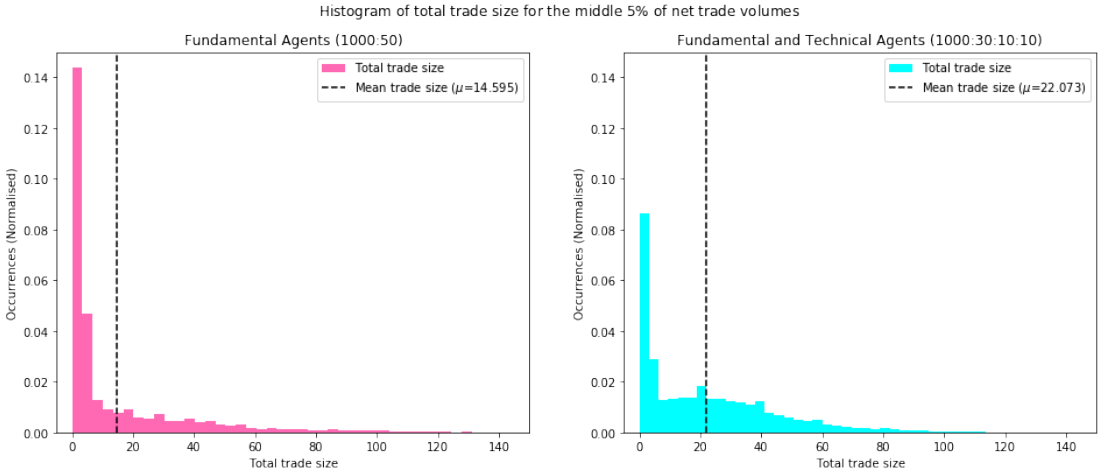

FIGURE 4.44: Histogram of total order volume for middle 5% of net trade volume in PRIME simulation (5 sec resample)

FIGURE 4.45: Histogram of price change for middle 5% of net trade volume in PRIME simulation (5 sec resample)

Section 4.2.1. As shown in Figure 4.46, a clear diminishing price impact of the trades can be observed, and its shape appears significantly smoother than the relationship observed in the simulation without technical agents (Figure 4.35). Whilst the discussion in this subsection has been qualitative, plotted results show that the addition of technical agents leads to more desirable initial impact behaviour.

4.6.6.2 Impact Reversion

Next, we revisit the analysis of reversion in market impact within the *PRIME* model after the inclusion of technical agents. As before, we observe a conditionality of the previous period order direction on the contemporaneous market impact (Figure 4.47), but this time to a clearer and a greater extent. In addition, the reversion could be observed

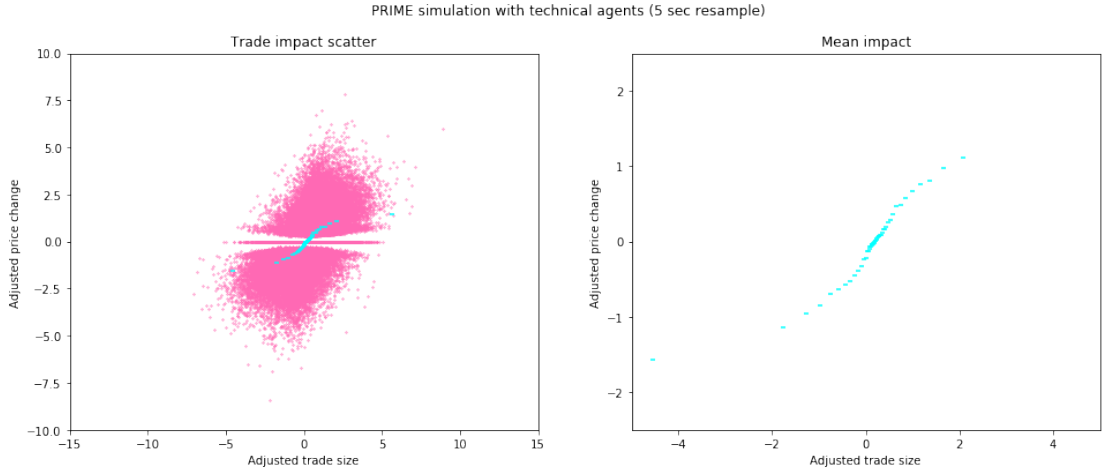

FIGURE 4.46: PRIME Market Impact (5 sec resample, adjusted)

around the small positive net adjusted trade sizes, that previously showed inconclusive results (Figure 4.36). On top of this, there is a small improvement regarding the decay in market impact (Figures 4.48 and 4.49). Despite the decay still not exhibiting the desired power-law behaviour after the first few lags, and instead showing a linear change, the lack of decay in market impact from the first time period to the second has now disappeared, even without changing the resampling period from 5 seconds to 10 seconds. Although this does not help us model the decay using power-law any better, the *PRIME* model still shows strong evidence of decaying market impact, even after the inclusion of technical agents.

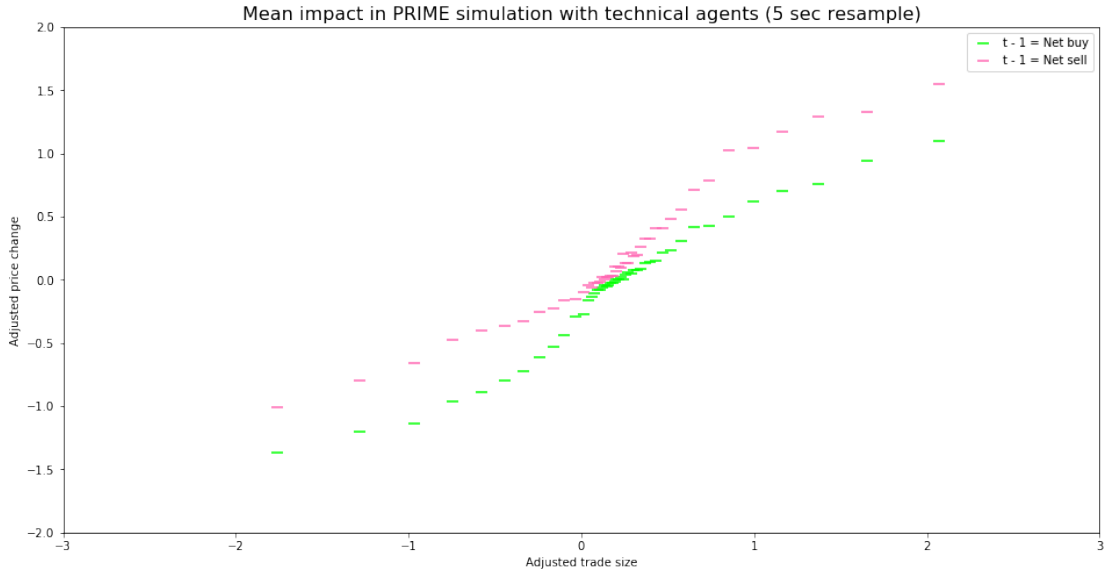

FIGURE 4.47: PRIME market impact coefficient over time (5 sec resample)

Overall, we believe the improvements in the clarity of impact reversion within our analysis originate from the improved representation of latent trade intentions within the

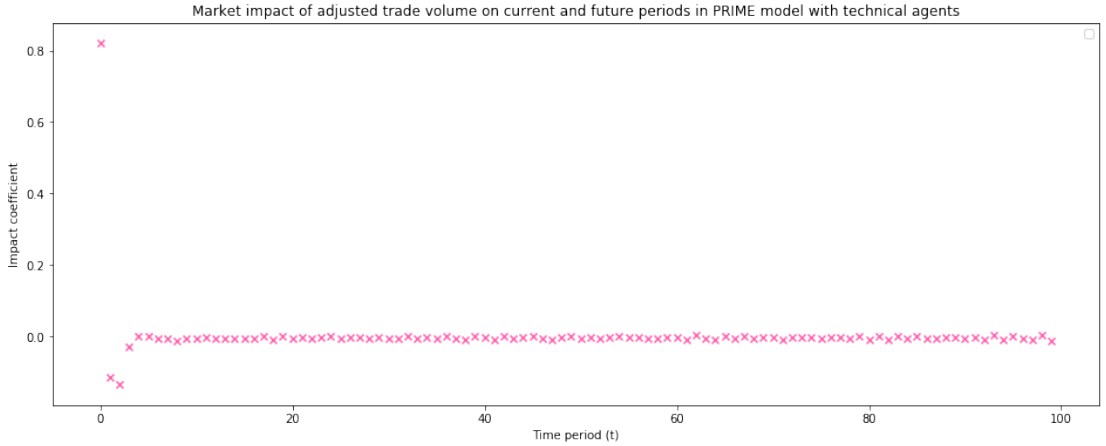

FIGURE 4.48: PRIME market impact coefficient over time (5 sec resample)

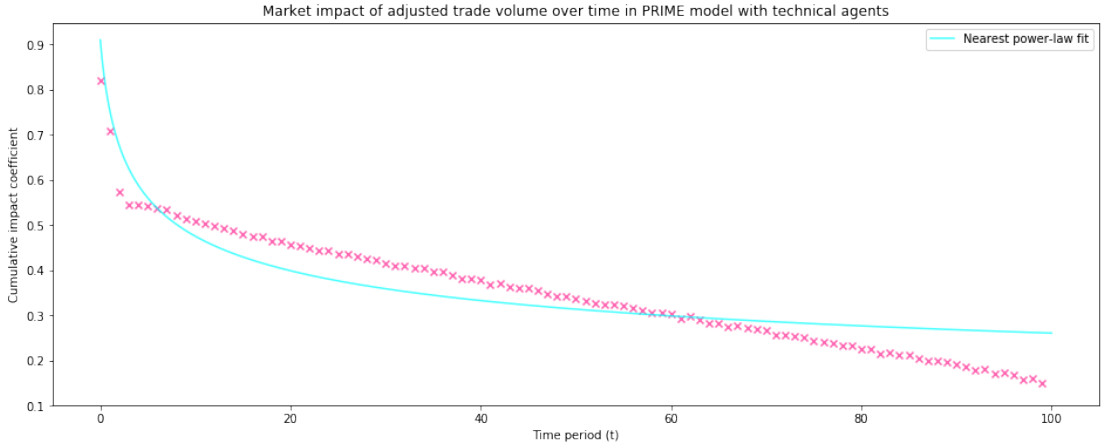

FIGURE 4.49: PRIME market impact decay over time (5 sec resample)

new simulation. Whilst the latent intentions of participants are not embedded explicitly into our agents, the mean-reversion agents provide the system with a behaviour that is somewhat similar to the latent intentions of market participants. Upon the price being pushed in a direction, these participants push the price in the opposite direction by consuming liquidity on the opposing side of the order book, as opposed to providing additional liquidity on the same side of the book.

4.6.6.3 Meta Orders

Finally, we take a look at the order sign autocorrelation upon introducing technical agents. Previously, the PRIME model showed a behaviour that is similar in form to the actual market data (Figure 4.41), but the magnitude and the decay in order sign autocorrelation were not quite in line with the market. When technical agents are added to the system, we now assume the ability to control this autocorrelation within the simulation by varying the ratio of technical agents. Unsurprisingly, the momentum agents

who place orders in the same direction as previous orders increase the order sign autocorrelation, whereas the mean reversion agents do the opposite and reduce the autocorrelation. This can be seen in Figure 4.50, where the configuration that removes the mean reversion agents (middle graph, Configuration: 1000 ZI-L, 40 ZI-M, 10 Momentum) exhibits higher autocorrelation, whereas removing the momentum agents (right graph, Configuration: 1000 ZI-L, 40 ZI-M, 10 Mean-reversion) returns lower autocorrelation when compared to the default technical simulation on the left (Configuration: 1000 ZI-L, 30 ZI-M, 10 Momentum, 10 Mean-reversion).

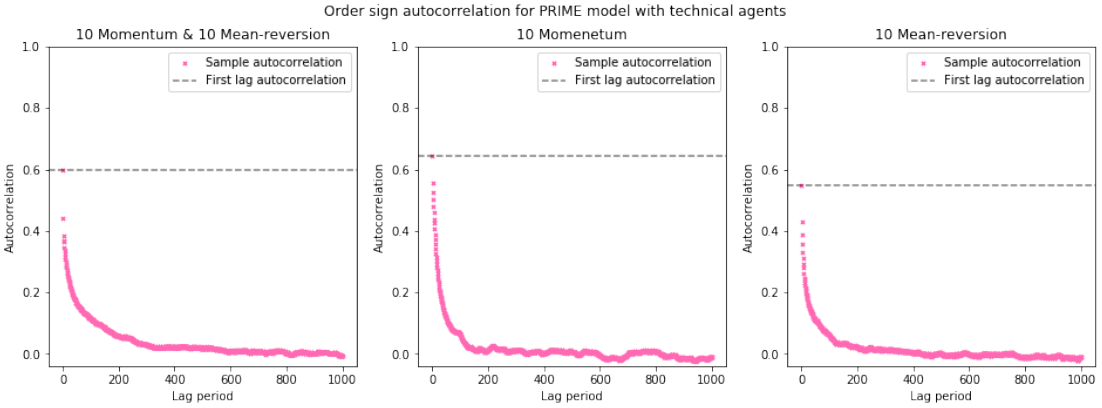

FIGURE 4.50: PRIME autocorrelation of Order Sign (Simulated Data, July 2020)

An interesting emergent behaviour of the system can be observed with the addition of technical agents (Figure 4.51). By adding a high ratio of momentum agents into the simulation (Middle graph, Configuration: 1000 ZI-L, 30 ZI-M, 20 Momentum), there is an overshoot of order sign autocorrelation, which is corrected over time. This is likely due to momentum agents pushing the simulation's mid-price too far away from the underlying price series, which over time the zero-intelligence market order agents act more aggressively in the opposite direction to correct the overshooting. Conversely, when there is a high ratio of mean-reversion agents in the simulation (right graph, Configuration: 1000 ZI-L, 30 ZI-M, 20 Mean-reversion), we observe an oscillation in order sign autocorrelation. This is likely due to the momentum agents overcorrecting any deviation from the mid-price, leading to a further correction in the opposite direction soon after. This behaviour suggests that the ratio of technical agents within the simulation should be limited, similarly to the restrictions imposed in Section 3.4.5, if the desired results are to be obtained.

4.6.7 Simulation configuration

The PRIME simulation environment is configured to replicate stylised microstructural behaviour using a heterogeneous population of agents. In our experiments, each simulation run corresponds to an eight-hour continuous trading session from 08:00 to 16:00

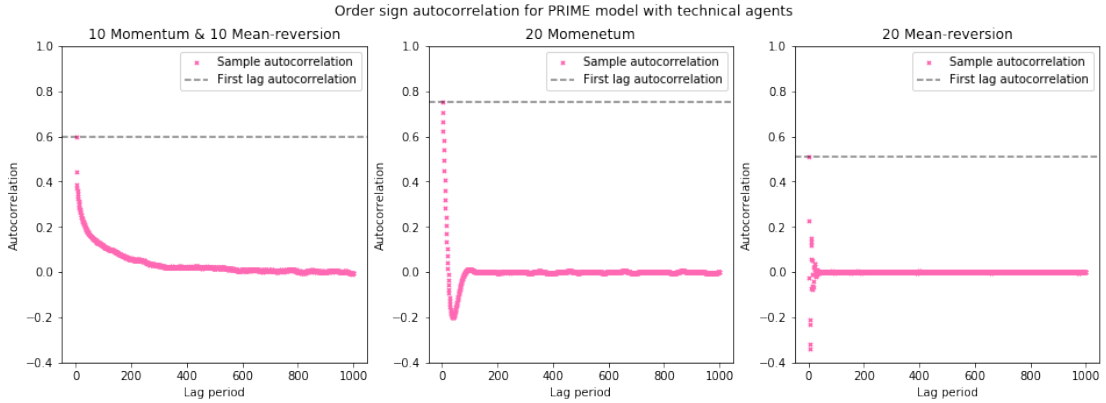

FIGURE 4.51: PRIME autocorrelation of Order Sign (Simulated Data, July 2020)

(this can be changed to the user's liking), during which agents operate asynchronously on the ABIDES platform.

To minimise financial friction, all agents are initialised with a cash balance of zero and may operate with negative balances. Trading occurs at a minimum resolution of 0.00001 BTC, and communication latency is set to zero, allowing for instantaneous message delivery between agents and the exchange. The initial order book is populated with a linearly increasing depth profile centred around the starting mid-price.

All agent types in PRIME interact with the market according to Poisson arrival processes, waking on average once per minute (i.e., $\lambda_a = 1/6 \times 10^9$). Market dynamics emerge from the interaction of four agent classes: ZI limit-order agents, ZI market-order agents, momentum agents, and mean-reversion agents.

The simulation includes 1,000 ZI-L agents that submit only limit orders priced as the mid-price plus uniform noise, and with a fixed order size of one unit. A further 30 ZI-M agents act as liquidity takers, submitting market orders with observation errors drawn uniformly from a specified range. These agents use a selective liquidity-taking (SLT) rule to determine trade size, governed by $V = \nu_0^{1-\Psi} V_{\text{top}}^{\Psi}$ with parameters $\nu_0 = 1$ and $\Psi = 0.5$.

To make the simulation more representative of the market, the setup includes technical agents (10 momentum and 10 mean-reversion). These agents also operate under Poisson wake-up schedules and submit market orders based on observed short-term and long-term price trends. Their order sizes follow a power-law drawn from a transformed Beta distribution: $U \sim \text{Beta}(3.5,\ 1)$, with final order size given by $\lceil 7/U \rceil$.

Table 4.1 summarises the full agent configuration used in the PRIME simulation framework.

TABLE 4.1: PRIME Simulation Agent Configuration

Component	Value / Description
Simulation-wide parameters	
Simulation duration	8 hours (08:00–16:00 session)
Oracle data source	Historic BTC/USD midprice from Binance
Tick size	0.00001 BTC
Initial LOB population	Linearly increasing depth around initial mid-price
Latency	Zero (all agent messages delivered instantly)
Inventory limit	1,000 units (applies to all agents)
Starting cash	0 (agents may run negative balances)
ZI Limit Order Agents (1,000 agents)	
Wake-up frequency	Poisson: avg. once every 60 seconds ($\lambda_a = 1/6 \times 10^9$)
Order type	Limit orders only
Order size	1 unit
Valuation	Mid-price + uniform noise $\epsilon \sim \mathcal{U}(-50,\ 50)$
ZI Market Order Agents (30 agents)	
Wake-up frequency	Poisson: avg. once every 60 seconds ($\lambda_a = 1/6 \times 10^9$)
Order type	Market orders only
Observation error	Uniform $\epsilon \sim \mathcal{U}(-20,\ 20)$
Order size	Dynamic (via SLT): $V = \nu_0^{1-\Psi} V_{\text{top}}^{\Psi}$
Selective Liquidity Taking	$\nu_0 = 1$, $\Psi = 0.5$
Momentum Agents (10 agents)	
Wake-up frequency	Poisson: avg. once every 60 seconds ($\lambda_a = 1/6 \times 10^9$)
Order type	Market orders based on trend
Order size	$\lceil 7/U \rceil$, where $U \sim \text{Beta}(3.5,\ 1)$
Short-term window	5 seconds
Long-term window	10 seconds
Mean-Reversion Agents (10 agents)	
Wake-up frequency	Poisson: avg. once every 60 seconds ($\lambda_a = 1/6 \times 10^9$)
Order type	Market orders based on reversal to trend
Order size	$\lceil 7/U \rceil$, where $U \sim \text{Beta}(3.5,\ 1)$
Short-term window	5 seconds
Long-term window	10 seconds

4.6.8 Model evaluation

Overall, the PRIME model offers an environment that can produce desirable market microstructural behaviour, whilst maintaining the ability to follow an underlying price series. It can also interact effectively with other agents in a realistic manner, without losing these properties. In addition, the inclusion of technical agents in the simulation not only increases the realism of the system from the perspective of introducing new representative market participants, but it was also shown to produce more desirable and controllable market behaviour (Section 4.6.6).

However, the model is unable to reproduce the functional form of impact reversion that decays following a power law in the real world, and is only able to reproduce reversion and meta-orders through indirect channels, as opposed to explicit agent behaviour such as latent trading intentions and division of market order volume. In addition, the model seems to have an upward bias in the net transactions (more buy volume than sell volume on average), as shown by the positively skewed trade size buckets in Figure 4.35. Although this final issue is likely due to the upward price drift of our underlying price series (BTC/USD, July 2020), the other shortcomings remain as areas to be improved.

In future iterations of PRIME, these issues should be addressed by attempting to incorporate latent liquidity into the system through the idea of more intelligent agents that often hide their intentions, similarly to the ZIP agent (Section 3.1.2), and incorporate a trading strategy such as VWAP or TWAP to limit the trade size for market order agents. Furthermore, the model should be taken one step further, from reproducing just the behaviour of the BTC/USD market to reproducing the same magnitude of market impact, reversion, as well as order sign autocorrelation. We believe this can be achieved in an explainable way within the PRIME model by adjusting the ratio of technical agents and the aggressiveness of participating agents.

4.7 Conclusion

In this chapter, we provided empirical evidence and analysis of three important microstructural features of the BTC/USD Futures market: market impact, impact reversion, and the autocorrelation of order signs. Our study reaffirmed the existence of the square-root law in the Bitcoin market post-2017 rally and contributed novel findings regarding both the reversion of market impact and the persistence in consecutive order flow direction.

From a simulation standpoint, our baseline model from Chapter 3 was insufficient to reproduce these interactive behaviours. We explored alternative methodologies such as zero-intelligence agents and the Santa Fe model as alternatives but found them lacking. These approaches were limited either in their ability to follow a specified price series or in their failure to generate order-sign autocorrelation.

To overcome these shortcomings, we developed a novel simulation framework, PRIME (Cho et al. (2023)), built on the ABIDES platform. PRIME features a heterogeneous population of agents designed to represent intuitive classes of market participants, such as fundamental and technical traders. This structure enables realistic endogenous interactions whilst maintaining the ability to track an exogenous price series. As a result, PRIME successfully reproduces several key empirical patterns observed in the cryptocurrency market, including the square-root law of market impact, impact decay, and the autocorrelation of order signs.

These contributions directly advance our first and third research objectives, with our analysis of stylised facts on market impact, reversion, and autocorrelation addressing the former objective by updating and adding to the empirical characterisation of the BTC/USD market, and our work on the PRIME framework answering the latter objective by incorporating market impact dynamics to the simulation. Although some limitations remain, PRIME represents, to our knowledge, the first crypto-specific simulation framework capable of reproducing the empirically observed market impact properties of a real exchange.

Chapter 5

Benchmarking Market-Makers

In this chapter, we present the experimental results associated with our fourth research objective of this thesis, namely the design and evaluation of reinforcement learning (RL) controlled market makers within the empirically grounded simulation environment developed in the preceding chapters. Specifically, this objective, outlined in Section 1.2, examines: the effectiveness of different RL architectures in learning to make markets within a realistic, interactive simulation environment; the influence of state-action space design on learning behaviour and overall performance; differences in performance across key market-making metrics, including inventory management, quoted spread, traded volume, and realised PnL; and the extent to which market-making models developed using static historical data translate to performance in a dynamic, multi-agent market simulation.

To answer these questions, we evaluate state-of-the-art agents drawn from the value-function, policy-gradient, and actor–critic families of reinforcement learning. These agents are trained and tested within the PRIME framework introduced in Chapter 4, allowing their behaviour to be assessed in the presence of realistic market impact and other stylised facts observed in the BTC/USD market. Finally, we benchmark the learned agents against a previous study (Spooner et al. (2018)), enabling a direct comparison between agents in an interactive simulation.

Based on prior findings in the reinforcement learning literature by Gašperov et al. (2021), we formulate the following hypotheses regarding the relative performance of different RL architectures in the market-making task. We hypothesise that DQN, as a value-based method, will exhibit fast and stable learning in simpler environments, but may struggle as the complexity of the state-action space increases. In contrast, we expect PPO, as a policy-gradient method, to scale more robustly to richer feature representations and denser reward structures, albeit at the cost of slower convergence. Finally, we hypothesise that A2C, as an actor-critic method, may achieve competitive performance when convergence occurs, but will be particularly sensitive to the stochasticity

of the simulation environment, potentially resulting in unstable learning or behaviour indistinguishable from random actions.

These hypotheses are evaluated empirically in the remainder of this chapter through an analysis of simulation results across varying levels of market complexity and feature richness, enabling a systematic assessment of each algorithm's effectiveness in the market-making task.

5.1 Recap of market making

As described in previous sections, market makers are participants in financial markets who provide liquidity by continuously quoting both buy and sell prices for an asset. In doing so, they facilitate transactions between temporally mismatched buyers and sellers, by holding inventory across time to bridge this gap. In return for bearing this inventory risk and offering liquidity to the market, market makers are compensated by quoting asymmetric prices, offering to buy at a lower price than they offer to sell. This is called the bid/ask spread (selling price less buying price) that the market-maker captures (or profits) every time they provide liquidity to two participants in opposing trade directions. The performance of a market maker ultimately depends on their ability to navigate the competing objectives inherent in this role: offering liquidity to the market whilst maximising risk-adjusted returns under varying market conditions. Profitability requires quoting a sufficiently wide bid−ask spread to generate margins, as well as maintaining a high transaction volume to monetise those margins. However, widening the spread can make the quotes less attractive to the counterparties, reducing trading activity. At the same time, managing market risk calls for minimising inventory exposure, but strict constraints on the inventory may limit trading opportunities and force the agent to unwind positions at unfavourable prices. A successful market maker must dynamically balance these trade-offs (spread size, trade frequency, and inventory risk) to consistently generate profit regardless of prevailing market conditions.

The market-making task outlined above can be naturally formulated as a sequential decision-making problem, and more specifically as a MDP. With the growing popularity of ML techniques, a number of researchers have turned to reinforcement learning as a potential solution to this problem. Most existing studies carried out so far adopt model-free RL techniques and train agents by interacting with historical price series. The key points of differentiation amongst these works typically lie in the choice of the RL algorithm architecture, the design of the state-action space, and the construction of the reward function. Gašperov et al. (2021) provide a high-level survey of research in this area, and their observations are generally in agreement with our own independent review of existing work. However, it remains difficult to make meaningful comparisons in previous studies. These studies are often evaluated using private datasets or

highly simplified environments, many of which make naive assumptions about market impact or ignore the effects of inter-agent interactions. This heterogeneity and lack of transparency in the experimental setup between researchers hinders reproducibility, obscures the relative strengths and weaknesses of each approach, and makes it challenging for future researchers to benchmark new models or build upon prior findings.

In our work, we address these limitations by training and evaluating three reinforcement learning-based market-making agents within a controlled, open-source multi-agent simulation environment designed to replicate the market microstructure of a cryptocurrency exchange, that we introduced previously (the PRIME framework). To ensure comparability, we standardise environmental randomness across agents by using the same random seed for the three agents and use the same state-action space architecture and reward function across experiments. Using this consistent interactive environment that captures realistic trading dynamics, we are able to fairly assess and report the relative performance of different RL algorithms under identical conditions. This setup enables meaningful and reproducible comparisons between agent designs and provides a set of results that can be used as a benchmark for future researchers in this space.

5.2 Recap of reinforcement learning

Reinforcement learning is the *third type* of machine learning alongside supervised and unsupervised learning, where the model learns to make optimal decisions through a process of trial and error, continuously interacting with and receiving feedback from its environment. This interaction environment is formulated using an MDP, which describes the world using sets of states and actions that evolve following a transition function, with a reward being attributed to each state-action pair via a reward function. The set of states describes all possible situations the decision-making agent can be in, or at least observe, and the actions represent all the choices this agent can make. Lastly, the policy function contains the probability of making an action at a given state, thus mapping the transition from one state to another. The traditional reinforcement learning method explores the state-action space through a trial-and-error process. Using a tabular representation of the state-action space, the agent learns the environment by combining exploration (usually a random action) and exploitation (usually the action with the highest expected reward) and attributes the received reward to the tabular state-action space. As the number of interactions increases, the agent eventually learns to interact with the environment optimally to maximise its rewards. However, as the environment in question becomes more complex, these tabular approaches become unusable, due to the curse of dimensionality leading to computational limitations. To get around this issue, modern reinforcement learning techniques use functions to approximate the state-action space. Whilst this strategy may not lead to the globally optimal solution, researchers have found that the approximation functions lead to impressive

performances across various applications by converging to what is likely a local optimum approximating the best potential decision.

5.3 Market-Maker Design

The most critical component of this chapter is the design of the market-making agent. In this section, we outline the process of constructing a market-maker that observes and acts upon a carefully selected subset of market information deemed most relevant for liquidity provision. The goal is to engineer a succinct and informative state-action space that retains essential features for market-making whilst omitting irrelevant or noisy signals. Effective feature engineering plays a central role in reducing the dimensionality of the learning space, thereby enabling the agent to converge toward a viable policy within a practical training horizon.

This convergence process forms the basis of our comparative study across different reinforcement learning architectures. We stress the importance of a well-engineered state-action space design, as poor feature engineering may result in slow convergence, convergence to a suboptimal local maximum constrained by the representational space, or even a complete learning failure. Such issues, in turn, may impair our experiment's ability to meaningfully differentiate between RL algorithms in terms of their suitability for the market-making task.

5.3.1 State Space Design

We begin by designing the state space observed by the market-making agent. The scope of information available to the agent is critical, as it directly impacts both the speed of learning and the limit of achievable performance. In theory, a larger state space capturing all elements of the environment allows the agent to converge toward a globally optimal policy. In our context, this would involve exposing the agent to the entire time series of Level-3 limit order book (LOB) data, including every market order, limit order, and cancellation placed by all participants. However, such an approach is computationally infeasible. Level-3 LOB data at nanosecond resolution is prohibitively large, and storing or processing it in real time within a simulation requires more computational capacity than is available even in modern high-performance computing clusters. Consequently, using such a high-dimensional state space would likely result in the agent failing to learn any meaningful strategy from its interactions, as the complexity of the environment would overwhelm its learning capabilities.

At the other extreme, selecting an overly coarse state space with too few features introduces a different set of problems. Whilst reducing the dimensionality of the input

space may accelerate learning and improve convergence speed, it often results in a poor representation of the environment. In such cases, the agent may converge to a suboptimal policy that is bounded well below the theoretical maximum. Worse still, the agent might learn behaviours that perform no better—or even worse—than random actions, as it is effectively being trained on data that are uninformative or misleading for the objective of optimal market-making.

To manage the trade-off between state spaces of differing resolution, we begin by constraining the feature set using domain knowledge. This ensures that the state space remains informative whilst avoiding unnecessary complexities that may hinder learning. The selected candidate features, which serve as inputs to the agent state-space, are as follows:

- Agent's current inventory
- Distance of the agent's best outstanding bid and ask quotes from the mid-price, expressed as a percentage of the spread (capped at 5)
- Current market bid-ask spread, capped at 20 ticks (maximum of 10 ticks per side, even when data is missing)
- Recent mid-price return, defined as the change in mid-price over the previous time step
- Order book imbalance at the top level: (bid volume − ask volume) / (bid volume + ask volume)
- Volatility of the mid-price over the past 20 observations
- Signed volume traded on the exchange since the agent's last wake-up

Whilst it is possible to include additional features, such as price and volume breakdowns across the top five levels of the order book or detailed statistics on order executions and cancellations, we chose not to expand the state space further. We believe that the selected feature set already forms a sufficiently rich representation of the market environment, and that it captures the core signals traditionally used by voice traders when making markets. Moreover, expanding the feature set would continuously increase the dimensionality of the state space, potentially impairing learning efficiency without a material gain in informational value.

We design the state space as a continuous set, rather than a discrete one, despite the underlying simulation operating on discretised representations of time, price, and volume. This choice reflects the fact that these discrete variables are a consequence of the simulation framework, rather than being intrinsic to the market-making task itself. In addition, to support the agent's learning process, all input features are normalised to

the range $[-1, 1]$, following standard practice in the RL market-making literature, as noted by Gašperov et al. (2021).

5.3.2 Action Space Design

The next step is to design the action space. In the market simulator, an agent must decide on the type of order it wants to submit, along with its size, price, and direction (e.g., submitting a limit order to buy 1000 units at \$100). However, allowing the agent to issue arbitrary combinations of orders across the entire price and volume spectrum (from zero up to some computational limit) would unnecessarily inflate the action space, significantly increasing the learning complexity. To mitigate this, we constrain the agent's interactions with the environment to a restricted set of action parameters. As with the state space, this decision involves a trade-off between expressiveness and tractability. Accordingly, the set of the agent's discrete action space is defined as follows:

- Place bid
- Place ask
- Place market order to buy
- Place market order to sell
- Cancel all outstanding bids
- Cancel all outstanding asks
- No action

The discreteness of the action space is consistent with previous work in this domain (Gašperov et al. (2021)). In our simulations, the agent is permitted to execute either a single action from this list or a constrained combination of actions (e.g., placing 1-unit limit orders to buy and sell simultaneously, both one tick away from the mid-price). These composite actions and the structure of the discrete action space are discussed in more detail later in this chapter.

5.3.3 Reward Function Design

Finally, we design the reward function to provide appropriate incentives for the agent to engage in genuine market-making activity. The primary goal of a market-maker is to generate profits, but another key function is to provide intertemporal liquidity to facilitate trades between buyers and sellers over time. To reflect this dual role, the agent's reward is constructed to promote profitability whilst discouraging speculative behaviour and encouraging sustained liquidity provision.

Profit is naturally included in the reward on a mark-to-market basis. However, profit alone is not sufficient. Without further constraints, the agent may adopt speculative strategies, such as taking directional bets or rapidly clearing inventory via market orders, thereby negating its role as a liquidity provider. To counteract this, we introduce a penalty for holding inventory, a standard technique also used in previous work (Gašperov et al., 2021, Section 4.7). However, the inventory penalty by itself may not be sufficient, as the agent may offload its position through aggressive market orders, which undermines its function as a net liquidity contributor. To address this, we impose an additional penalty proportional to the volume executed via market orders, discouraging reactive position-clearing that drains liquidity from the market. Furthermore, to ensure continuous presence of the market-maker in the order book, we include a quote proximity penalty based on how close the agent's best bids and asks are to the mid-price, with maximum penalties applied when either side of the book is unquoted.

Taken together, our reward function is comprised of four components: (i) profitability, (ii) inventory holding, (iii) quote placement relative to the mid-price, and (iv) use of market orders. The functional form is given in Equation 5.1. Note that we multiply the profit per trade element of our reward by 100 to match the magnitude of the penalties we impose.

$$\text{Reward}_t = \left(\frac{P_t}{V_t} \cdot 100\right) - I_t - (\phi_b + \phi_a) - \mu \tag{5.1}$$

$$P_t = \text{cash}_t + h_t \cdot m_t - \text{cash}_0 \quad \text{(mark-to-market profit)}$$

$$I_t = |h_t| \quad \text{(inventory penalty)}$$

$$\phi_b = m_t - B_t \quad \text{(mid-price to best bid, capped at 10)}$$

$$\phi_a = A_t - m_t \quad \text{(mid-price to best ask, capped at 10)}$$

$$\mu = \begin{cases} v_t, & \text{if action = market order} \\ 0, & \text{otherwise} \end{cases} \quad \text{(penalty for market order usage)}$$

$$V_t = \text{cumulative traded volume up to time } t$$

where:

- h_t: inventory holdings at time t,
- m_t: mid-price at time t,
- B_t: agent's best bid quote,
- A_t: agent's best ask quote,

- v_t: volume of last market order executed by the agent.

We acknowledge that this reward function could be further refined. For example, incorporating a risk-adjusted return metric based on prevailing market volatility might yield a more nuanced measure of agent profitability. However, such refinements, while potentially beneficial, are unlikely to materially affect the agent's ability to make markets, therefore is unlikely to lead to a difference in convergence behaviour of different RL algorithms. Therefore for the purposes of this comparative study, we prioritise incentive structures aligned with the foundational responsibilities of a market-making agent (profitability, inventory control, passive liquidity provision, and minimisation of aggressive order usage) to observe emergent behaviours of each market-making agent and their ability to learn from the multi-agent market environment.

5.4 Simulation Setup

As with Chapters 3 and 4, our choice of simulation environment is the ABIDES platform. In addition to the benefits discussed previously, researchers at JP Morgan have created an OpenAI Gym environment for the ABIDES platform (Amrouni et al. (2021)). This allows for a more convenient implementation of reinforcement learning algorithms in the ABIDES platform. As part of our contribution to the research community, we were able to augment the environment to follow our novel model (PRIME, Chapter 4), rather than defaulting to a mean-reverting environment available in the original ABIDES codebase. This allows RL agents to train and compete in a controlled environment that takes into account known market properties that are not incorporated in other simulations, thus providing a testing environment more representative of the real markets.

5.4.1 Market-making agents

The reinforcement-learning architectures evaluated for the market-making task in this chapter are Deep Q-Networks (DQN, Mnih et al. (2015)), Proximal Policy Optimisation (PPO, Schulman et al. (2017)), and Advantage Actor–Critic (A2C, Mnih et al. (2016)). These algorithms were selected to represent three distinct methodologies in reinforcement learning: value-based methods, policy-gradient methods, and actor–critic methods, respectively. Together, they provide a representative comparison of contemporary approaches to sequential decision-making in stochastic environments.

DQN, a value-based off-policy algorithm, is designed for discrete action spaces and benefits from enhancements such as experience replay and target networks. These features are expected to contribute to early training stability and fast convergence in our

setting. Given the bounded and normalised state space and frequent trading, we anticipate that DQN will perform well in the simpler scenarios. However, as we increase the dimensionality of the state or expand the discrete action set (e.g., by introducing finer price levels or quoting strategies), DQN may become less effective. The curse of dimensionality makes it difficult to populate an informative replay buffer, and the lack of generalisation between similar states may slow down learning or lead to instability.

On the other hand, PPO, a policy-gradient algorithm with clipped updates, is well-suited to environments with noisy rewards and moderately large action spaces. Its robustness to variance in policy updates and its ability to optimise stochastic policies make it a strong candidate for the market making task, especially as we increase the environment's complexity. Unlike DQN, PPO does not rely on replay buffers and instead optimises the policy directly, allowing it to generalise better in state spaces where similar inputs may lead to different optimal actions. While PPO is typically slower to converge due to its on-policy nature, we expect it to outperform DQN as the dimensionality and behavioural nuances of the environment increase.

Lastly, the actor-critic algorithm, A2C, offers a hybrid between value estimation and direct policy learning. It is more sample-efficient than pure policy-gradient methods, and in environments with structured feedback and frequent rewards like ours, it can achieve strong performance. Like PPO, A2C is likely to scale better than DQN as the state-action space expands. However, its on-policy nature, lack of experience replay, and sensitivity to initialisation may lead to noisy convergence or delayed learning. In practice, we may observe A2C stagnating in early training before undergoing "breakthroughs", and we may also not be able to reliably make these "breakthroughs", as A2C is very dependent on the initial condition of the learning algorithms.

5.4.2 Benchmark model: Spooner *et al.* (2018)

In addition to the market-making agents that we stated above, we also evaluate a previous study by Spooner et al. (2018) (Section 2.3; hereafter, Spooner model). This method adopted a tile-coded linear TD-learning agent for market making. Like us, they focused on the design of state and action spaces, as well as introducing a reward function that discourages speculative behaviour.

The state comprises three continuous features, each discretised into 10 uniform bins over a fixed range. For example, the inventory is capped at ± 50 and divided such that the values from -50 to -40 fall into bin 0, -40 to -30 into bin 1, and so on. This discretisation supports tile coding, allowing efficient generalisation in a linear value function approximation setting:

- **Inventory**: the agent's current inventory position.

- **Spread**: the prevailing bid-ask spread.
- **Mid-price Movement**: change in the mid-price since the last timestep.

The value function is then expressed as a weighted combination of multiple tile codings:

$$\hat{v}(x) = \sum_{i=0}^{N-1} \lambda_i \hat{v}_i(x) = \sum_{i=0}^{N-1} \lambda_i \sum_{j=0}^{n_i-1} b_{ij}(x) w_{ij} \tag{5.2}$$

where:

- N is the number of tile codings,
- λ_i is the weight for tile coding i, with $\sum_i \lambda_i = 1$,
- $b_{ij}(x) \in \{0, 1\}$ indicates if bin j in tile coding i is active,
- w_{ij} is the learnable weight associated with that bin.

The action space consists of 10 discrete actions:

- **Actions 0–8**: place symmetric limit orders at increasing distances from the mid-price.
- **Action 9**: submit a market order to reduce inventory, defined as:

$$a_9(t) = -\alpha \cdot \text{Inv}(t) \tag{5.3}$$

The base reward function includes mark-to-market and execution PnLs:

$$\Psi(t_i) = \psi_a(t_i) + \psi_b(t_i) + \text{Inv}(t_i) \cdot \Delta m(t_i) \tag{5.4}$$

To penalise inventory that moves against the market, the final reward includes an asymmetric damping term:

$$r_i = \Psi(t_i) - \max(0, \eta \cdot \text{Inv}(t_i) \cdot \Delta m(t_i)) \tag{5.5}$$

In our implementation, we set $\eta = 0.3$, consistent with the best-performing configuration reported in the original paper. For the market order action, we set $\alpha = 1$, corresponding to a full liquidation of the current inventory, since we did not test partial clearing strategies.

5.4.3 Simulation Length

In training our reinforcement learning agents within the PRIME framework (Table 5.1), another key design decision involves time. More specifically, we are referring to the duration of each training session and the frequency with which the market-making agent observes the market to decide whether or not to act.

Training an agent in a single continuous market session until convergence may appear efficient, but doing so risks overfitting to the idiosyncrasies of that particular run. This is especially problematic given that the RL agents actively influence the environment, creating path dependencies. To mitigate this, the simulation must run long enough for the market to stabilize and for the agent to discover its stable behaviour patterns, while also resetting initial conditions across batches to avoid dependence on a specific trajectory.

Equally important is the agent's decision frequency, or wake-up rate. The agent must act frequently enough to respond meaningfully to changes in the environment but not so frequently that it unnecessarily observes a state that has not changed since the previous wake-up, which will lead to a lack of learning efficiency and simulation scalability.

In light of these considerations, we define a single training episode as a one-hour market session. Within each episode, the market-making agent is scheduled to wake up 100,000 times, or once every 36 milliseconds. This frequency ensures that the agent observes and responds to nearly every market change. For reference, the rest of the market consists of around 1,000 agents, each of which wakes up once per minute. As such, the market-making agent is activated almost twice as often as the entire population of background agents combined. This provides ample opportunity to observe changes in the LOB and make informed decisions in response to evolving market conditions.

5.4.4 Hyperparameters and Experimental Settings

To conclude the simulation setup, we summarise the key environment configurations, training protocol (Table 5.2), and reinforcement-learning hyperparameters used throughout the experiments in this chapter (Tables 5.3, 5.4, 5.5). All reinforcement-learning agents are implemented using the Stable-Baselines3 library. These settings are reported to ensure full reproducibility of the results presented in Section 5.5 and to allow future researchers to build on this work.

Table 5.1: Simulation environment configuration (PRIME framework)

Parameter	Value
Simulator	ABIDES (OpenAI Gym interface)
Market model	PRIME (Chapter 4)
Market mechanism	Continuous double auction
Underlying price process	Exogenously specified (price-following enabled)
Market-impact model	Endogenous (PRIME)
Background agent population	$\approx$ 1,000 heterogeneous agents
Background agent wake-up rate	$\approx$ one action per minute per agent

Table 5.2: Training protocol and episode design

Parameter	Value
Episode duration	1 simulated hour
Decision steps per episode	100,000
Market-maker wake-up interval	$\approx$ 36 ms (simulated time)
Training batches	100
Timesteps per batch	100,000

Table 5.3: DQN hyperparameters used in Chapter 5 experiments

Hyperparameter	Value
Learning rate	1×10^{-4}
Discount factor γ	0.95
Replay buffer size	1,000
Batch size	64
Target network update interval	1,000

Table 5.4: PPO hyperparameters used in Chapter 5 experiments

Hyperparameter	Value
Learning rate	1×10^{-4}
Discount factor γ	0.95
Batch size	64
Rollout length (`n_steps`)	2,048
Optimisation epochs per update (`n_epochs`)	10
Entropy coefficient (`ent_coef`)	0.01

5.5 Simulation Results

We train RL agents with various combinations of state and action spaces with progressively increasing complexity, as disussed in Section 5.4. Our goal is to understand how these design choices influence the behaviour and performance of the market-making agent, and which RL architecture is most suitable for the task of market-making. To evaluate this, we track various metrics throughout the course of training. These include:

- **PnL per volume**: a normalised profit measure, representing the agent's mark-to-market profit per unit of volume traded.

TABLE 5.5: A2C hyperparameters used in Chapter 5 experiments

Hyperparameter	Value
Learning rate	5×10^{-5}
Discount factor γ	0.99
Rollout length (`n_steps`)	20
Entropy coefficient (`ent_coef`)	0.005
Value function coefficient (`vf_coef`)	0.5
Max gradient norm (`max_grad_norm`)	0.5
RMSProp optimisation (`use_rms_prop`)	True
Normalise advantage (`normalize_advantage`)	True

- **Average spread**: the average distance between the agent's outstanding bid and ask quotes, capturing how competitively the agent is quoting.
- **Average inventory**: a proxy for inventory risk, reflecting the agent's typical exposure to directional price movement.
- **Volume traded per wake-up**: a measure of how active the agent is during each decision period.
- **Reward**: the actual reward signal fed back to the agent, incorporating profit and various penalties.

We use these metrics to assess how well the agent is making markets, not only in terms of profitability, but also in how closely it mimics the desirable behaviour of a liquidity provider: quoting competitively, maintaining low directional exposure, and retaining market presence. The results we discuss in the section below will compare agent performance across different reinforcement learning algorithms and training configurations, providing insight into how state-action design impacts emergent market-making behaviour.

Note that each agent was trained for approximately five million episodes per simulation. We found this training horizon sufficient to observe either convergence or stagnation in performance across all experimental configurations. To reduce the impact of initial transient states and better capture long-run behaviour, we compute a moving average of the rewards over a window of 100,000 steps (in line with a single market session). This helps mitigate biases introduced by early-stage fluctuations before agents settle into equilibrium performance within a given market session. Furthermore, depending on the nature of each tracked metric, we apply different smoothing techniques: for noisier variables such as inventory and reward, we use a moving median; for all other metrics that are more discrete, a moving mean is used.

5.5.1 Varying the state-action space design

5.5.1.1 Compact State-Action Space Simulation

The first simulation we run employs a compact state-action space to establish a baseline for agent performance under constrained informational and strategic settings. The agent observes a low-dimensional market representation consisting of four normalised features: (i) the prevailing market spread, (ii) the agent's inventory level, (iii) the mid-price return from the previous to the current timestep, and (iv) the order book imbalance between the best bid and ask volumes. Each feature is clipped and scaled to lie within the $[-1, 1]$ range to promote numerical stability and reduce sensitivity to outliers. This representation provides minimal but carefully selected information about the prevailing market dynamics, allowing the agent to respond to high-level signals while minimising the risk of overfitting to noise.

The action space is also constrained to five discrete choices, reflecting a simplified but expressive set of market-making behaviours. At each decision point, the agent may: (1) execute a market order to fully clear its inventory, (2) cancel all outstanding limit orders, (3) place a limit bid one tick below the mid-price, (4) place a limit ask one tick above the mid-price, or (5) do nothing. These actions allow the agent to participate passively or actively, reduce risk, or abstain from action altogether. By training agents within this compact regime, we investigate whether rudimentary market-making behaviours, such as spread quoting and inventory management, can emerge from sparse state inputs and limited tactical flexibility.

The results are revealing. Although all agents exhibit significant noise in their profit-per-volume trajectories, the DQN agent achieves a predominantly positive outcome, while the performance of PPO and A2C remains less conclusive (Figure 5.1). However, the elevated magnitude and volatility of DQN's profitability suggest that profits are primarily driven by directional inventory exposure rather than consistent spread capture. This interpretation is supported by the inventory dynamics shown in Figure 5.2, where all agents (most notably DQN) maintain sustained directional positions, indicating speculative rather than liquidity providing behaviour.

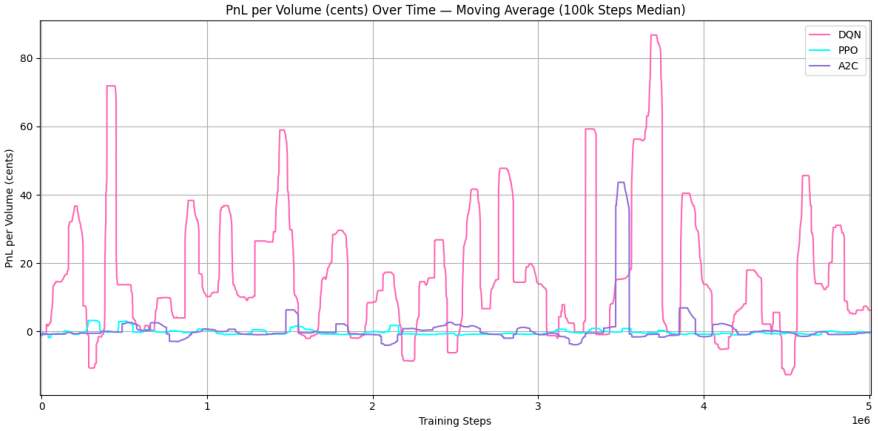

FIGURE 5.1: PnL per volume over time (compact state-action space setup with 4 observable states and 5 discrete actions)

FIGURE 5.2: Inventory levels over time (compact state-action space setup with 4 observable states and 5 discrete actions)

Further evidence of the breakdown in market-making behaviour is found in the average spread trajectory. Throughout most of the training, both DQN and A2C maintained quotes on only one side of the book (either bid or ask), but rarely both. Although PPO performed marginally better, none of the agents consistently posted two-sided liquidity. This can be seen in Figure 5.3, where all agents tend to incur spread penalties close to 10 ticks, which implies the absence of a competitive quote on one side. A penalty of 20 would correspond to the absence of quotes on both sides; a penalty of 10 usually indicates the agent is quoting in only one direction (although in theory, the spread could be constantly around 10 units wide, we confirmed that this was not the case). These results highlight that the agents are not acting as liquidity providers in the traditional sense.

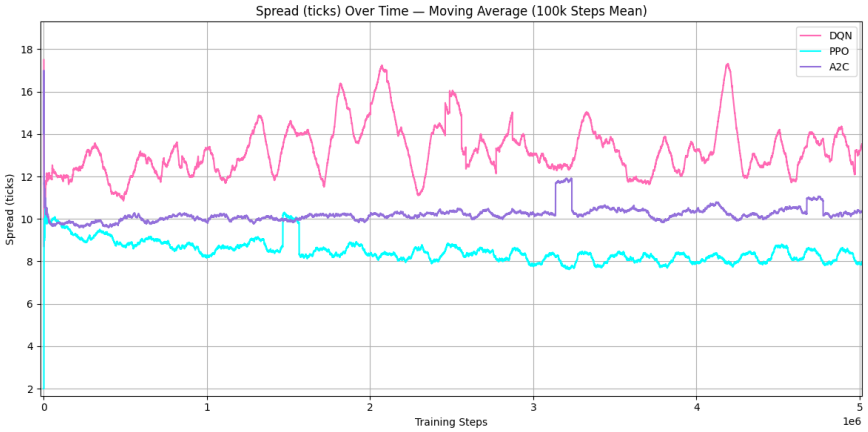

FIGURE 5.3: Average quoted spread (compact state-action space setup with 4 observable states and 5 discrete actions)

Interestingly, agent behaviour diverges more clearly when considering trading activity, measured by volume traded per wake-up. As seen in Figure 5.4, DQN exhibits the highest overall volume but with substantial volatility, suggesting erratic engagement with the market. PPO, by contrast, gradually increases its trading activity, eventually matching DQN in volume but with greater stability. A2C trades the least and demonstrates a modest decrease in activity over time, suggesting that it may have learned a more risk-averse or inert policy. However, in the context of this compact set-up, this higher traded volume does not correspond to an improved market-making ability, but rather a more sporadic directional position taking.

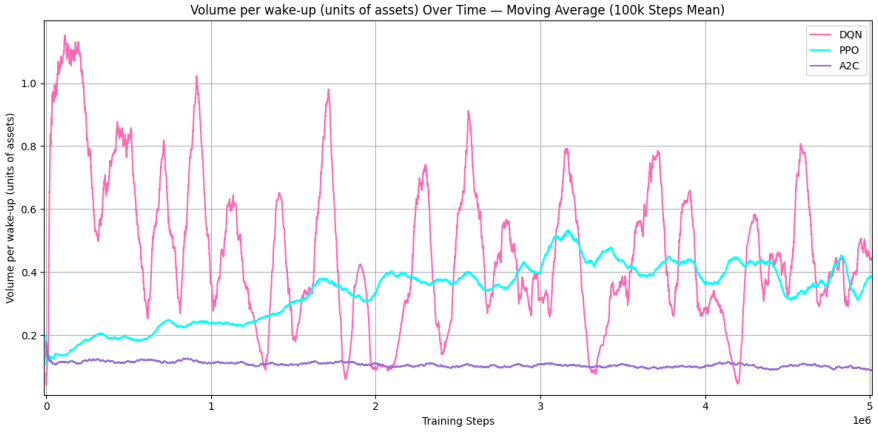

FIGURE 5.4: Average volume traded per wake-up over time (compact state-action space setup with 4 observable states and 5 discrete actions)

Finally, Figure 5.5 illustrates the evolution of the reward over time. Since rewards are dominated by mark-to-market PnL, their trajectories mirror the noisy behaviour observed in profitability. None of the agents demonstrate a stable learning pattern in this configuration, with the reward signals remaining highly volatile throughout. In

summary, this compact simulation reveals that whilst agents can occasionally exploit directional price movements for gain, they fail to exhibit robust, liquidity-providing strategies under this minimal setup. These findings motivate the exploration of richer state representations and more expressive action spaces in subsequent experiments.

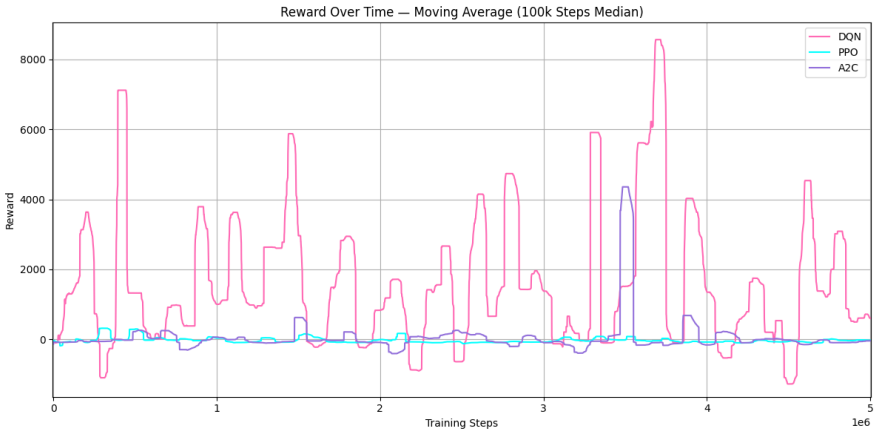

FIGURE 5.5: Reward over time (compact state-action space setup with 4 observable states and 5 discrete actions)

5.5.1.2 Extended State-Space Simulation

Building on the limitations identified in the compact configuration, we now extend the agent's observation space to include more granular microstructural features that may enhance its ability to act as a market maker. Whilst the previous setup relied on just four features (spread, inventory, mid-price return, and order book imbalance), the extended configuration expands this to eight distinct inputs. Specifically, the agent now additionally observes: (i) the relative distance of the agent's own bid and ask quotes from the mid-price, (ii) short-term price volatility computed over the last 20 mid-price observations, and (iii) the total traded volume since the agent's previous wake-up. As before, all features are normalised and clipped within the $[-1, 1]$ range to ensure numerical stability during training. This richer representation provides the agent with additional market context and feedback on its own quoting behaviour, which should allow the agent to infer additional relevant information from the market to help with its market-making activity. In turn, we are looking to evaluate whether incorporating these microstructural signals enables agents to carry out market-making behaviours such as tight spreads, consistent two-sided quoting, and improved inventory control relative to the more abstract setup discussed earlier.

Turning first to the inventory dynamics (Figure 5.6), a noticeable improvement is observed in the DQN agent. Although it initially displays high inventory levels, the agent progressively learns to reduce its exposure over time, albeit with intermittent spikes.

Both PPO and A2C also show more controlled inventory paths compared to their performance in Figure 5.2, with PPO equilibrating at a slightly elevated inventory level compared to others. This level may be consistent with the behaviour of a passive market maker who accumulates small positions whilst providing liquidity passively.

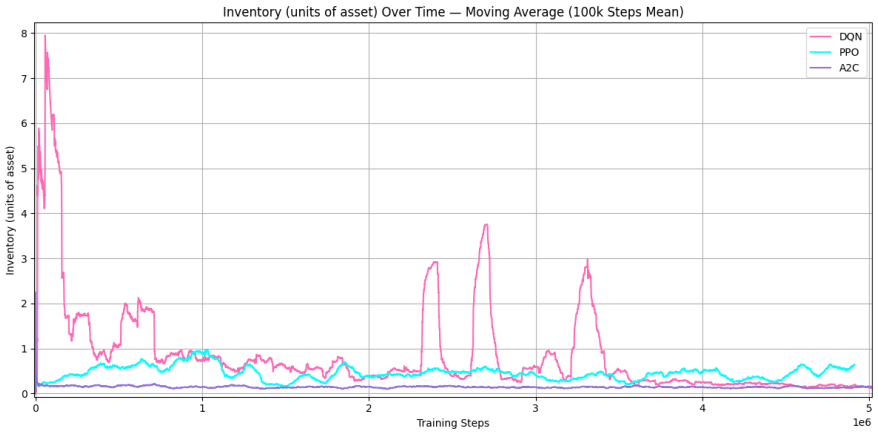

FIGURE 5.6: Inventory levels over time (extended state-space setup with 8 observable states and 5 discrete actions)

Looking at the spread (Figure 5.7), DQN and A2C still exhibit a one-sided quoting behaviour, failing to consistently offer both bids and asks. In contrast, the PPO agent demonstrates a marked improvement. It successfully learns to place two-sided quotes, thereby fulfilling a key function of a market maker, and also learns to tighten the spread as it learns from the environment. This behavioural change is likely related to the higher and more stable inventory seen in Figure 5.6. The ability to maintain a tighter spread also enables the PPO agent to support higher trading volumes over time (Figure 5.8), indicating a greater level of liquidity provision to the market. Unlike in the compact setup, the increase in trading volume here is more plausibly a consequence of effective spread management than directional speculation.

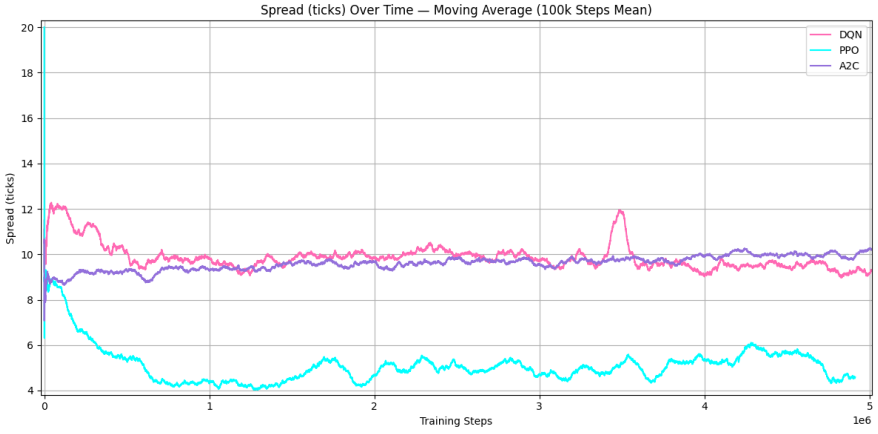

FIGURE 5.7: Average quoted spread (extended state-space setup with 8 observable states and 5 discrete actions)

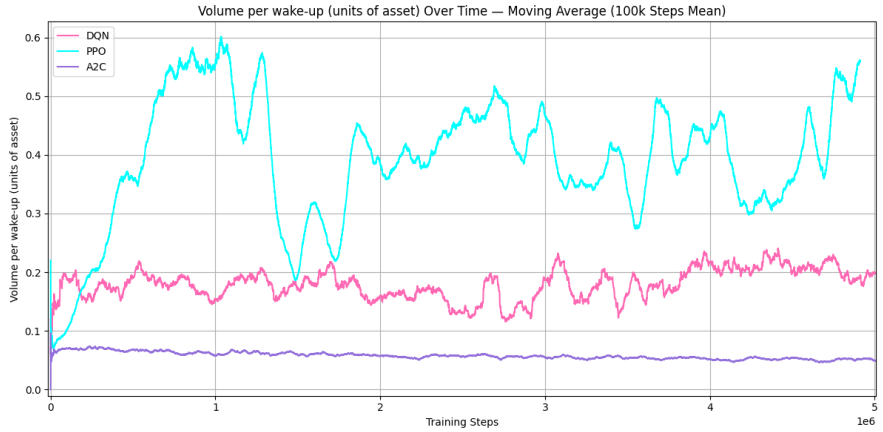

FIGURE 5.8: Average volume traded per wake-up over time (extended state-space setup with 8 observable states and 5 discrete actions)

Despite these improvements in quoting and participation, none of the agents were consistently profitable. Of the three, PPO incurred the smallest losses of approximately 0.3 cents per unit traded, while also demonstrating superior consistency. Nonetheless, the DQN agent does show some learning progress, reducing its losses over time, although it remains vulnerable to sharp drawdowns linked to periods of high inventory holdings. As shown in Figure 5.10, the reward trajectories largely mirror the mark-to-market profit patterns, but PPO stands out once again due to its tighter spread management and disciplined inventory control, both of which contribute positively to its overall reward.

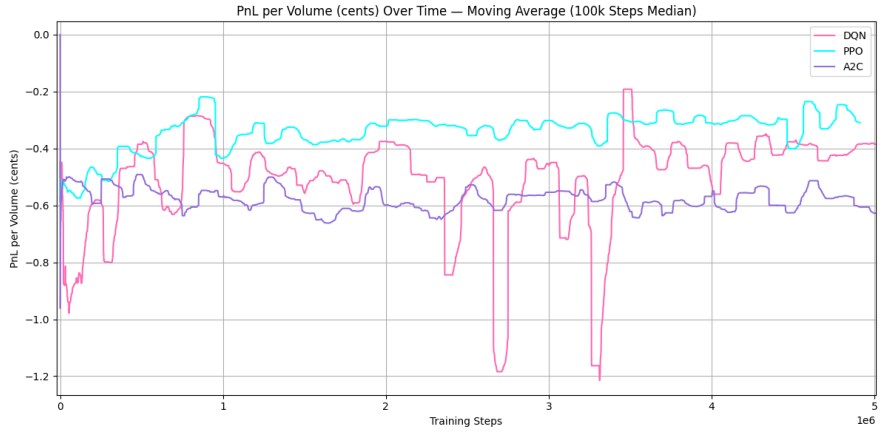

FIGURE 5.9: PnL per volume over time (extended state-space setup with 8 observable states and 5 discrete actions)

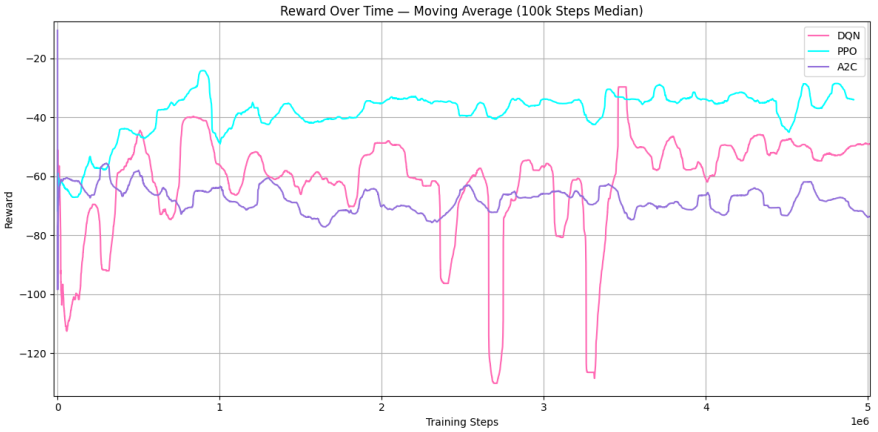

FIGURE 5.10: Reward over time (extended state-space setup with 8 observable states and 5 discrete actions)

5.5.1.3 Adjusted Action-Space Simulation

In this configuration, we restructure the agent's action space, not by hugely increasing its size but fundamentally altering its design to encourage more authentic market-making behaviour. Earlier results (with the exception of PPO in the extended state space setup) showed that agents often failed to quote on both sides of the book, undermining their role as liquidity providers. To address this, we modify the action set such that any time a limit order is placed, it must be accompanied by a corresponding quote on the opposite side.

The agent now selects from six discrete actions: submitting a market order to fully clear its inventory; cancelling all outstanding limit orders; placing symmetric limit orders by quoting one unit each at two ticks below and above the mid-price; choosing between two asymmetric quoting strategies either with a tighter bid (one tick below mid) and wider ask (two ticks above), or the reverse; or finally to choose to take no action at all. This structure ensures that the agent actively engages with the market as a liquidity provider rather than exploiting directional price trends. Whilst the action space remains compact, it is explicitly constructed to enforce two-sided quoting, yet flexible enough to accommodate different quoting styles. This setup allows us to test whether a constrained but well-engineered action space can induce consistent and effective market-making behaviour, particularly when combined with the enriched state space developed earlier.

Turning first to inventory management (Figure 5.11), we observe that although the DQN agent initially struggles with controlling its exposure to the market, it gradually converges towards a level of inventory stability similar to that of PPO and A2C agents. This suggests that all three agents learned to reduce their directional risk over time, likely through a mix of skewed quoting and strategic market orders, demonstrating a behaviour that is closer to a market-maker.

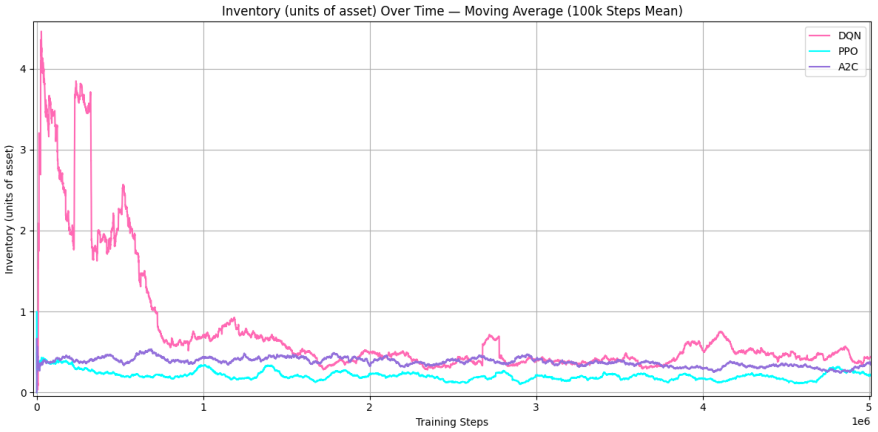

FIGURE 5.11: Inventory levels over time (adjusted action-space setup with 8 observable states and 6 discrete actions)

Examining the quoted spreads (Figure 5.12), we find that both PPO and A2C maintain tight spreads around three ticks, while the DQN agent starts off around six and converges to a wider spread of approximately four to five ticks. Although this might suggest a less effective market-making strategy by DQN, the agent's higher trading volume (Figure 5.13) indicates otherwise. Over the course of training, DQN learns to consistently capture this wider spread, implying that it executes more trades at favourable prices and does so more frequently. This hypothesis is confirmed when examining profitability: DQN exhibits the highest PnL per unit traded (Figure 5.14), demonstrating that it is capable of effective liquidity provision despite quoting wider spreads. In other words, DQN is balancing the conflicting priorities of being a market-maker more effectively than its peers. Nevertheless, all three agent types were still unable to make a profit the market simulation.

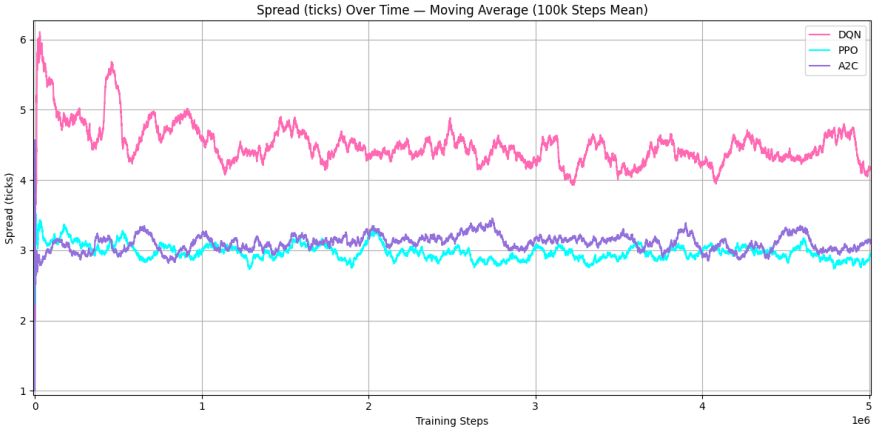

FIGURE 5.12: Average quoted spread (adjusted action-space setup with 8 observable states and 6 discrete actions)

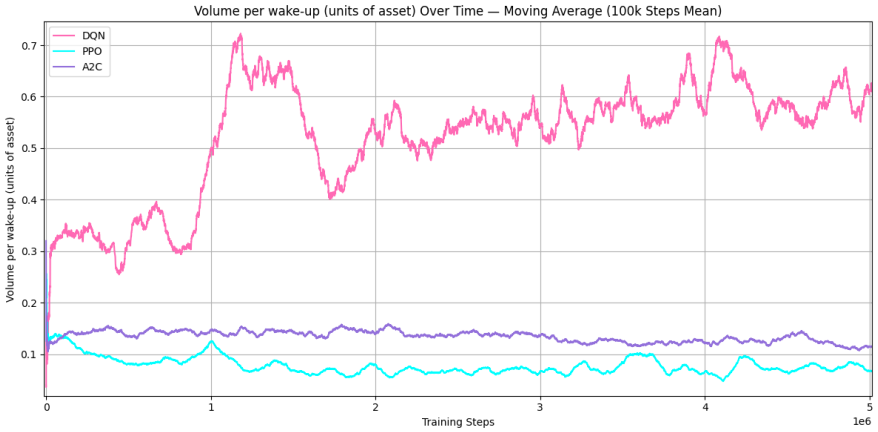

FIGURE 5.13: Average volume traded per wake-up over time (adjusted action-space setup with 8 observable states and 6 discrete actions)

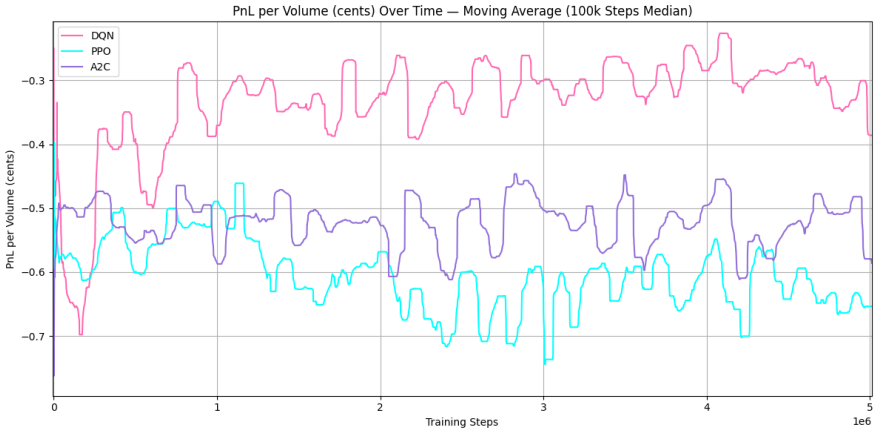

FIGURE 5.14: PnL per volume over time (adjusted action-space setup with 8 observable states and 6 discrete actions)

Reward trajectories (Figure 5.15) again largely mirror the profit profiles, with DQN outperforming its competitors. Notably, PPO performs worse under the adjusted action-space setup than in the previous configuration. In the extended state-space setup prior to action-space engineering, PPO achieved a reward of approximately -30 and a per-trade loss of about 0.3 cents. However, with the enforced market-making structure, its loss per trade increases to 0.7 cents and its reward drops to around -80. This contrast suggests that PPO may benefit more from the unconstrained exploration of the action space, whereas the structured design appears to aid DQN and A2C, especially the former, which now rivals PPO's previous performance.

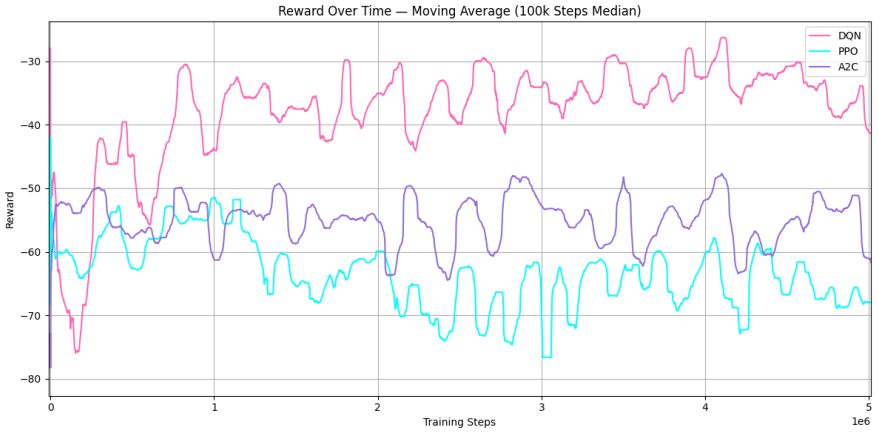

FIGURE 5.15: Reward over time (adjusted action-space setup with 8 observable states and 6 discrete actions)

5.5.1.4 Full State-Action Space Simulation

In our final experimental configuration, we implement a comprehensive state-action framework that provides agents with the most detailed market representation and the most expressive action space tested so far. This setup is designed to assess whether reinforcement learning (RL) agents can develop more effective and robust market-making strategies when given a highly flexible trading environment.

The state space remains consistent with the eight features introduced in the extended state-space setup. This decision is motivated by the strong performance of the PPO agent under that configuration, and the subsequent decline in its performance when the action space was constrained in the adjusted action-space setup. This suggests that the state representation was already sufficiently rich for effective learning (at least for PPO). Until other agents can match or exceed PPO's performance from Figure 5.10, the state space is unlikely to be the limiting factor.

In contrast, the action space is significantly expanded to comprise ten discrete actions, allowing more granular quoting strategies. These actions are:

1. Cancel all outstanding limit orders.
2. Submit a market order to flatten existing inventory (buy if short, sell if long).
3. Place symmetric limit orders one tick from the mid-price (bid at mid -1, ask at mid $+1$).
4. Place symmetric limit orders two ticks from the mid-price (bid at mid -2, ask at mid $+2$).

5. Place symmetric limit orders three ticks from the mid-price (bid at mid -3, ask at mid $+3$).
6. Skew quotes with a tighter bid and wider ask (bid at mid -1, ask at mid $+2$).
7. Skew quotes with a wider bid and tighter ask (bid at mid -2, ask at mid $+1$).
8. Place a wide bid and moderately wide ask (bid at mid -3, ask at mid $+2$).
9. Place a moderately wide bid and wide ask (bid at mid -2, ask at mid $+3$).
10. Take no action.

This diverse set of actions enables agents to fine-tune both the symmetry and aggressiveness of their quotes in response to current market conditions and their own inventory profiles, supporting a wider range of adaptive market-making behaviours.

Again turning first to inventory exposure (Figure 5.16), we observe a similar dynamic to the adjusted action-space setup. The DQN agent begins with large swings in inventory but learns to manage its exposure over time. However, both its initial and final inventory levels are higher in this configuration than before, and the A2C agent converges to an equilibrium inventory slightly higher than its counterparts. spread behaviour follows the same pattern: DQN stabilises around 4 to 5 ticks, while PPO and A2C maintain tighter spreads near 3 ticks (Figure 5.17).

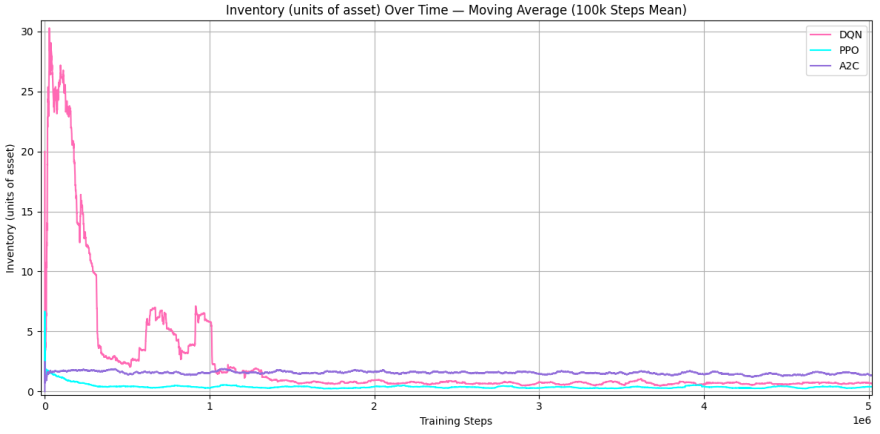

FIGURE 5.16: Inventory levels over time (full state-action space with 8 observable states and 10 discrete actions)

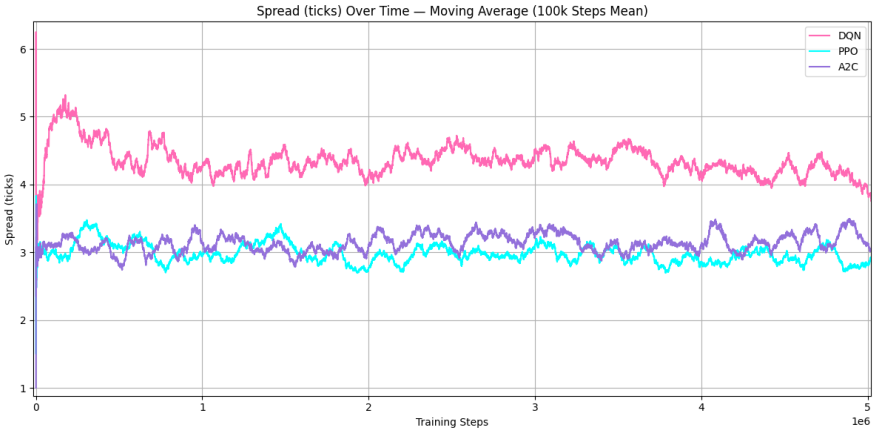

FIGURE 5.17: Average quoted spread (full state-action space with 8 observable states and 10 discrete actions)

Trade volume trends also resemble earlier experiments, with DQN agent increasing its traded volume over time, and PPO reducing its traded volume (Figure 5.18). What is slightly different is the level at which traded volumes settle. Both DQN and A2C have learned to trade more volume than before, whereas PPO has maintained its low market presence.

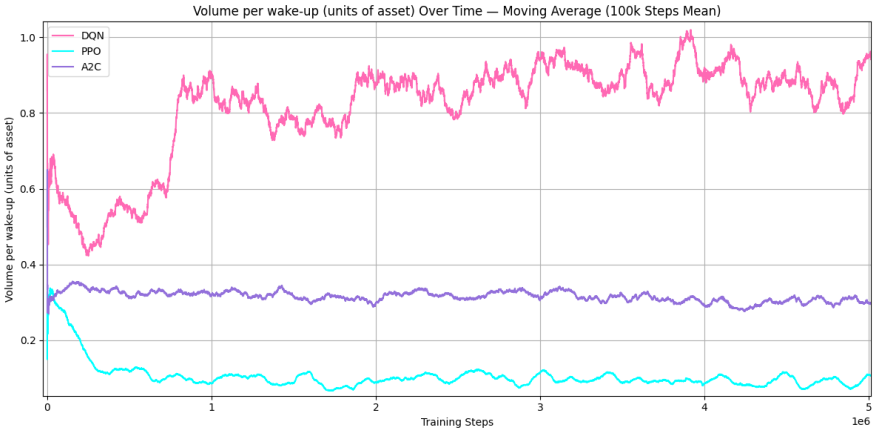

FIGURE 5.18: Average volume traded per wake-up over time (full state-action space with 8 observable states and 10 discrete actions)

PnL and reward outcomes again mirror prior results (Figures 5.19 & 5.20), with one key exception: while DQN and A2C see little performance gain, the PPO agent improves markedly. Its average reward increases by approximately 10 points, driven largely by enhanced profitability. This suggests that the additional flexibility in the action space was particularly beneficial for PPO, allowing it to capitalise on the detailed state information more effectively than in previous configurations. To our surprise however, the agents were not able become profitable despite having access to quoting wider and skewed limit orders. Furthermore, this result is still far below PPO agent's performance when agent actions were not constrained to be market-making.

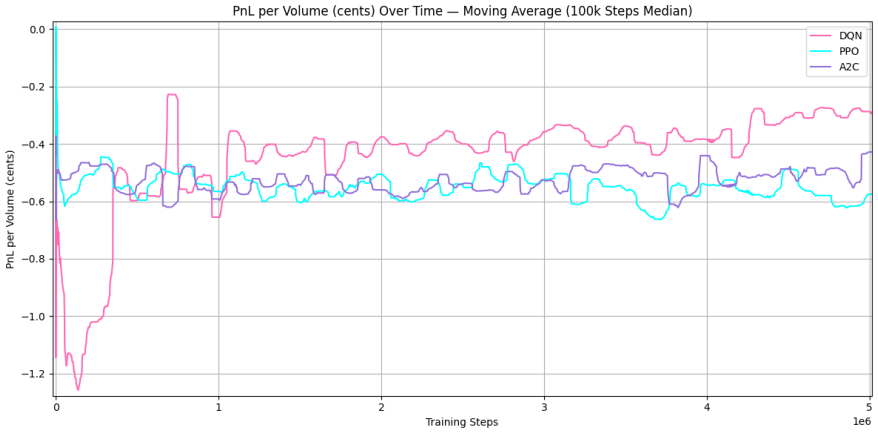

FIGURE 5.19: PnL per volume over time (full state-action space with 8 observable states and 10 discrete actions)

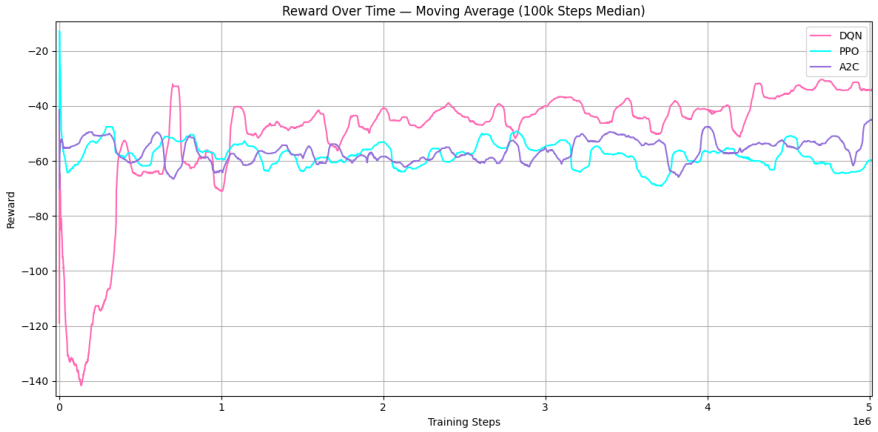

FIGURE 5.20: Reward over time (full state-action space with 8 observable states and 10 discrete actions)

5.5.1.5 Evaluation of training results

Across the four configurations, our experiments revealed interesting relationships between the complexity of the state-action space and the emergence of effective market-making behaviour in reinforcement learning agents. In the compact configuration, all agents struggled to fulfil the role of liquidity providers. Although DQN exhibited the highest profitability, this outcome was largely attributed to directional speculation rather than systematic spread capture. Supporting evidence includes persistent inventory imbalances, minimal two-sided quoting, and elevated spread penalties. The agents' rewards were noisy and unstable, with no clear learning signal, underscoring the insufficiency of a minimal state-action space to induce genuine market-making behaviour.

Expanding the state representation while keeping the action space fixed led to marked improvements. The PPO agent, in particular, learned to quote on both sides of the book,

tightened its spreads, and maintained a relatively stable inventory, all of which are hallmarks of successful liquidity provision. Although no agent achieved consistent profitability, losses were reduced and trading behaviours became more structured. These results suggest that the enriched state-space including features such as quote distances, recent volatility, and signed traded volume enabled agents to make more informed and market-aware decisions.

In contrast, restructuring the action space to enforce two-sided quoting had mixed effects. Although the developed action space encouraged more realistic market-making policies and constrained harmful speculation, it reduced flexibility. As a result, PPO's performance deteriorated, while DQN benefited significantly, learning to balance wider spreads with higher execution rates to generate improved profitability. These findings highlight that not all algorithms respond equally to structured constraints: PPO appeared to rely more on flexibility in action selection, while DQN capitalised on the engineered features and the choices provided.

The final configuration, which combines the extended state space with a fully expressive action space, produced mixed results between agents. Although all agent performance increased following the addition of more actions, PPO was unable to reach the high level of rewards that it achieved using a more flexible action space design. This suggests that while structural constraints encourage liquidity provision, they may inhibit agents that otherwise excel at discovering profitable strategies under freer regimes.

As for agents not being profitable across all simulations, we have several hypotheses. First, it may be due to the constrained action space allowing the placement of limit orders too narrow for the prevailing volatility of the simulation. Second, the market environment may be trending with low volatility, which creates unfavourable conditions for market makers, forcing them to liquidate positions as mark-to-market losses accumulate. Third, the inventory penalty may be too large, discouraging agents from holding positions long enough to complete round-trip trades profitably. Lastly, it is possible that the agents have not yet fully converged. We believe the first or second explanation is the most likely cause, as training performance has plateaued across most metrics across all simulations after one or two million steps (and therefore ruling out incomplete training). Furthermore, agents tend to maintain different levels of inventory equilibrium, suggesting that the inventory penalty is not the main issue. In addition, the fact that PPO achieved better results when unconstrained also supports the hypothesis that the action space is overly restrictive. In future experiments, we intend to incorporate an adjusted profitability metric that uses a "baseline" maximum profit that can be generated from capturing the spread to better understand whether it was the environment or the agent that led to constant loss-making.

In summary, our results from training the agents demonstrate that both state and action space design are critical to shaping market-making behaviour in RL agents, but the effects of tweaking the state and the action space vary depending on the RL architecture used. While compact representation with insufficient information about the market led to speculative behaviour, enriched observations and carefully structured actions promote more disciplined and effective liquidity provision. Agent-specific sensitivities to action-space constraints also suggest that architecture choice should be matched to the degree of structural bias and design introduced into the learning environment.

5.5.2 Evaluation of trained models

Using the trained market-making agents we have developed thus far, we evaluate and compare the market-making ability of the best-performing models from each architecture: DQN and A2C from the full state-action space configuration (Section 5.5.1.4), and PPO from the extended state-action space model (Section 5.5.1.2). In addition, we include a comparison with the market-making agent developed by Spooner et al. (2018), one of the earlier and most frequently referenced works in the domain of RL market making. While this work was briefly introduced in Section 2.2.2.2, we detail our implementation of their methodology in the following section.

5.5.2.1 Empirical results

After training the Spooner model for 5 million steps using the same PRIME configuration as our market-makers, we evaluate all agents over 1000 market-making sessions, each lasting 12 minutes (20,000 steps at a 36 ms wake-up frequency). To ensure comparability, we report only the central 90% of the results in all plots throughout this section, as extreme outliers tend to distort the scale and hinder effective comparison between agents.

Focusing first on inventory management (Figure 5.21), we observe that the DQN, PPO, and A2C agents demonstrate behaviour consistent with the final phase of their respective training simulations, with PPO being the most prudent with its inventory exposure. Interestingly, the Spooner model exhibits a much wider range of inventory outcomes, yet still achieves a lower average inventory than the A2C agent.

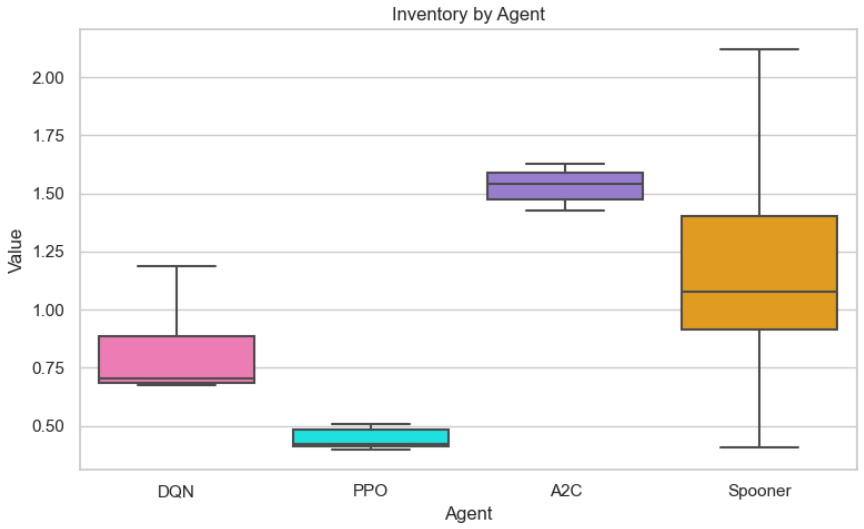

FIGURE 5.21: Distribution of mean inventory held by each agent across 1000 12-minute market sessions (plot shows median, middle 50% and middle 90% of data)

Next, we examine the PnL per trade achieved by each agent (Figure 5.22). None of our agents were able to consistently generate profits from market making during training, and this trend persisted during the batches of 12-minute market sessions. The Spooner model also failed to achieve profitability in the PRIME environment on average, and it performed worse than all of our agents in terms of median PnL. However, unlike our agents, the Spooner model exhibited a heavier right tail in its PnL distribution, with profitable outcomes observed for around the upper quartile of results.

FIGURE 5.22: Distribution of final PnL per trade across 1000 12-minute market sessions (plot shows median, middle 50% and middle 90% of data)

Lastly, we look at the agents' average bid/ask spread across market sessions (Figure 5.23). We find that A2C consistently maintained the tightest spreads, with wider spreads observed for DQN and PPO, in a pattern again consistent with observations from the later stages of training we saw in sections 5.5.1.4 and 5.5.1.2. The Spooner model maintained a notably wider spread than all of our agents during its sessions, but did so consistently, indicating a stable but wider quoting behaviour that the strategy has equilibrated towards.

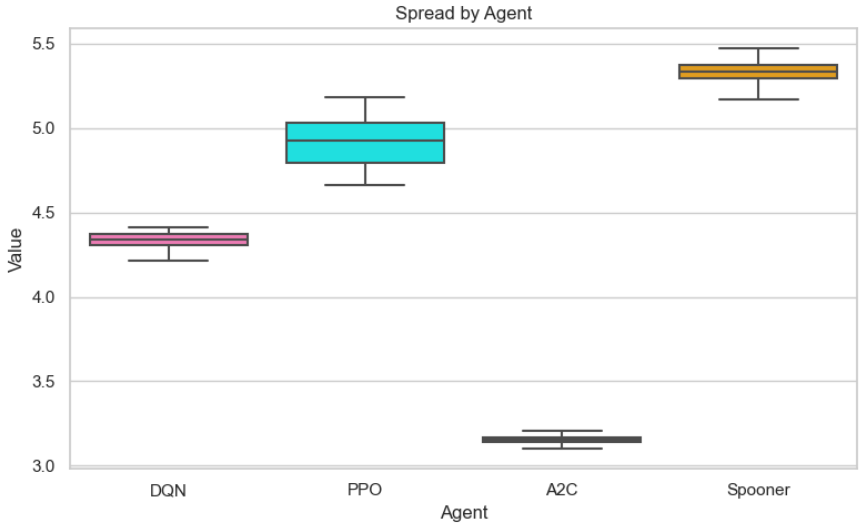

FIGURE 5.23: Distribution of mean spreads by each agent across 1000 12-minute market sessions (plot shows median, middle 50% and middle 90% of data)

5.5.3 Discussion

The empirical results across our evaluation sessions broadly reflect the behaviours observed at the end of training for each reinforcement learning model, as discussed in Sections 5.5.1.4 and 5.5.1.2.

Amongst our agent configuration, PPO appears to be the most promising candidate for market making. Although it did not achieve profitability during testing, its ability to reduce inventory exposure effectively whilst incurring the smallest losses suggests that the agent is able to learn how to market-make from the environment with limited feature engineering. We believe that with a more expressive or continuous action space, PPO may be able to further refine its quoting behaviour and move closer to profitability whilst maintaining its ability to provide liquidity.

The feature-engineered DQN model also shows potential. Its structured action space has enabled it to adopt a consistent quoting policy, which again was not profitable but was able to balance the spreads and the inventory it showed to the market in a fairly

consistent manner. We expect further feature engineering to improve the performance of this methodology.

By contrast, the A2C model clearly struggled to learn a viable market-making policy. It exhibited limited responsiveness in quoting or inventory management. However, it is noteworthy that the A2C agent produced relatively tight distributions across most evaluation metric except PnL, which suggests that the engineered state-action features constrained its behaviour in a consistent, albeit suboptimal, manner. This indicates that while the model did not learn to make effective decisions, the provided state-action space nonetheless anchored its outcomes within a stable regime.

The Spooner agent produced a notably wider range of outcomes across all metrics. This may be attributed to its tile coding architecture, which coarsely partitions the state space and introduces greater sensitivity to small shifts in state values. Despite this, the model consistently maintained a competitive quoting spread without any explicit penalty for wide spreads, indicating that their state-action design embedded useful priors about optimal quoting behaviour. Given that the Spooner action space included wider quoting distances than our agents and that it settled on a higher equilibrium spread, this could be suggesting that the wider spreads may be closer to the global optimum, and that our agents may benefit from the option to post wider bids and asks to the LOB.

Finally, while the Spooner model was able to generate profits in the upper quartile of its outcomes, its average performance remained significantly lower than that of our agents —including the underperforming A2C. This suggests that the coarse discretisation of the state space in the Spooner model may be insufficient for effective market-making in a more realistic and dynamic environment such as our PRIME simulation framework. In contrast, our approach that is centred on carefully engineered continuous state features and the use of function approximators is able to yield more consistent performance across market sessions. These findings underscore the importance of RL agent architecture and state-space design when applying reinforcement learning to the market-making problem.

5.6 Conclusion

This chapter set out to address the research questions associated with our fourth research objective concerning the effectiveness of reinforcement-learning approaches to market making in a realistic, interactive simulation environment. In particular, we examined: (i) how different RL architectures perform when learning to make markets, (ii) how variations in state-action space design influence learning behaviour and performance, (iii) how agent performance varies across key market-making metrics, and

(iv) the extent to which models developed on static historical data translate to performance in a dynamic, multi-agent market. The findings of this chapter are summarised below with respect to each of these questions.

To investigate these questions, we applied reinforcement learning (RL) to the market-making problem using a realistic agent-based simulation environment. Leveraging the PRIME framework developed previously, we designed a series of experiments to evaluate three RL algorithms (DQN, PPO, and A2C) across progressively more complex state-action spaces. Each experiment tested the agent's ability to learn profitable and disciplined liquidity-provision strategies in a setting shaped by realistic market microstructure.

Our initial hypotheses were fairly supported by the empirical results from our training, though with several important caveats. As expected, DQN excelled in low-complexity environments. It was the only agent to achieve a positive reward under compact state-action settings, albeit through speculative rather than liquidity-providing behaviour. Additionally, feature engineering, particularly the imposition of structural constraints requiring two-sided quoting, had a disproportionately positive effect on DQN, highlighting its dependence on a well-defined and discrete action space. These findings reinforce the notion that value-based methods can be effective in constrained environments when guided by informed domain priors.

PPO, by contrast, demonstrated great robustness in a more complex state-space environment with a freer action space design. It was the only agent to consistently adopt two-sided quoting behaviour and maintain tight spreads when the action space was left unconstrained. This supports our hypothesis that PPO's policy-gradient architecture is well suited to exploiting richer state representations, particularly in noisy reward settings where stochastic policies can benefit. However, its performance declined when structural constraints were imposed on the action space, suggesting that PPO's strength lies in its capacity for broad and flexible policy exploration. In the final full state-action experiment, PPO partially recovered its advantage, confirming both its scalability and its sensitivity to the design of the action space.

A2C, in contrast, failed to exhibit meaningful learning across all configurations. Despite its theoretical sample efficiency, it struggled to achieve stable policy updates in the face of environmental noise and high variance. Its occasional behavioural divergence appeared stochastic rather than learned, reinforcing our initial concerns about the algorithm's fragility in complex multi-agent financial simulation, especially without additional stabilisation techniques such as entropy regularisation or bootstrapped exploration.

When compared against the Spooner model, our agents combining feature engineering with ML-based function approximators demonstrated superior overall performance. This was particularly evident in the more stable and consistent outcomes across key

market-making metrics such as PnL, inventory, and spread. In contrast, the Spooner model exhibited a much wider dispersion in results, suggesting less reliable behaviour, likely originating from its coarser state-space representation designed for a static market without consideration of market impact or feedback. Nonetheless, some of the design choices in Spooner *et al.*, such as the dampened reward formulation that penalises speculative profits from adverse market movements, and the inclusion of market orders of varying magnitudes, offer valuable insights that could be incorporated into our models to improve their performance.

Taken together, these findings yield some key insights. First, the design of the action space plays a critical role in shaping learning outcomes, sometimes even more so than the design of the state space, depending on the choice of RL architecture. Second, while PPO appears best suited to high-dimensional and dynamic environments, DQN remains competitive when paired with effective feature engineering. By contrast, A2C, at least in its standard form, appears ill-suited for the market-making problem. More broadly, our results highlight the practical challenges of deploying RL in realistic market environments. Even when agents exhibit signs of learning, their behaviours often fail to generalise or stabilise in richer simulations, and converge to unprofitable strategies. This issue was not unique to our agents, and also appeared in existing methods trained on historic data such as the Spooner model.

Through our work in this chapter, we provide a benchmark framework for future researchers to test and compare their RL market-making agents. This framework enables controlled, apples-to-apples comparisons of different learning algorithms and design choices. Although we only evaluated one existing method alongside our agents, the framework is extensible and can be used to assess a wide range of RL market-making models in consistent and realistic settings.

Future work extending the progress made in this chapter should begin by exploring larger and less constrained action spaces, especially to further test the capabilities of policy-gradient methods. On top of this, additional feature engineering should be pursued to identify key state variables that may significantly enhance learning, particularly in configurations that benefit both value-based and policy-gradient algorithms. Lastly, the reward structure warrants further investigation to determine whether it can be refined to better incentivise profitable liquidity provision, particularly in terms of tuning the balance between the agent PnL and the ratio of penalties related to inventory risk, order aggressiveness, and spread capture, perhaps by using a similar technique used by Spooner et al. (2018).

Chapter 6

Conclusions and Future Work

This research set out to address four research objectives. The first was to extract and analyse stylised facts from the BTC/USD market, with a focus on statistical regularities observed across returns, order flow, and trading volume. The second objective was to develop a systematic methodology for tuning a multi-agent simulation on the ABIDES platform such that it reproduces the stylised facts observed on a real cryptocurrency exchange. The third objective focused on incorporating market impact dynamics into this simulation in order to model realistic trading costs, whilst retaining the simulation's ability to follow an underlying price series. Finally, the research aimed to design and evaluate RL-controlled market makers within this simulated environment, with the goal of understanding how different RL architectures and state-action space designs affect key market-making performance metrics.

6.1 Research Outcome

The first research objective was addressed in Chapters 3 and 4. In these chapters, we collected and analysed high-quality market microstructure data from the BTC/USD market using the API of the world's largest cryptocurrency exchange, Binance. The richness and granularity of this dataset enabled us to verify well-established stylised facts from traditional asset classes, such as volatility clustering and the absence of linear autocorrelation in returns, whilst also uncovering empirical characteristics specific to cryptocurrency markets.

Building on this empirical foundation, we introduced a novel simulation tuning methodology that fits a response surface to observed market data in order to calibrate a market

simulation composed of simple agents, thereby addressing the second research objective. This approach enables the reproduction of key real-world dynamics without reliance on complex or opaque stochastic processes. Using this same agent-based framework, we further addressed the third research objective by reproducing realistic market impact phenomena that arise endogenously through agent interaction. The framework presented in Chapter 4, PRIME, not only captures market impact upon order execution, but also reproduces the reversion and autocorrelation properties of market orders observed in empirical data. As a result, transaction costs and slippage that traders face are embedded directly within the simulated environment, allowing market participants to interact under realistic trading conditions without the need for external impact models. To our knowledge, this is the first interactive multi-agent simulation framework for cryptocurrency exchanges that is capable of reproducing realistic market impact dynamics in this manner.

In the final part of the research (Chapter 5), we investigated the design and evaluation of market makers trained via reinforcement learning (RL) within the realistic, microstructure-aware environment developed in earlier chapters. Specifically, we examined how different RL algorithms, namely DQN, PPO, and A2C, respond to variations in state-action space design and structural constraints. To this end, RL agents were deployed in progressively more complex simulation settings, yielding several key insights. First, the results highlight the importance of state-action space design. DQN performed best in constrained environments, benefitting from structured feature engineering and rigid action templates that mitigate known limitations of value function based methods. In contrast, PPO excelled in higher dimensional and less restricted settings, demonstrating strong adaptability to complexity but suffering performance degradation when structural constraints limited its exploratory capacity, particularly within the action space. Across all configurations, A2C failed to achieve stable or meaningful learning, underscoring its sensitivity to noise and instability in multi-agent market environments without additional regularisation. When benchmarked against the framework proposed by Spooner et al. (2018), the agents developed in this thesis consistently outperformed the existing model within the more realistic PRIME environment. Overall, these results yield two important insights for the RL market-making literature. First, state-action space design can exert a greater influence on performance than state representation, though this effect is architecture-dependent. Second, while policy-gradient methods such as PPO demonstrate strong performance in complex environments, value-based approaches such as DQN remain highly competitive when paired with carefully engineered features and constrained action spaces. This work therefore establishes a benchmark for evaluating RL-based market makers in a realistic exchange environment and provides a foundation for future comparative studies.

In conclusion, this thesis presents a robust and flexible testing ground for future research into cryptocurrency market microstructure. By decoupling empirical investigation from the limitations of historical data and opaque market impact models, the proposed framework enables transparent, reproducible experimentation within a realistic simulated exchange. As a first demonstration of its capabilities, the environment was applied to the study of RL-based market making, yielding insights into the interaction between algorithm choice, state-action space design, and market realism. The potential applications of this framework extend well beyond the experiments presented here, and it is hoped that this work will support further research into agent-based modelling, market impact, and algorithmic trading in cryptocurrency markets.

6.2 Future Work

We envision future work proceeding in two main directions. The first involves improving the multi-agent model of the financial exchange to address several limitations identified in this research. For example, certain missing stylised facts highlighted in Chapter 3, such as the distribution of order sizes or the detailed shape of the volume–volatility correlation, could be better replicated through enhanced agent behaviour or order flow modelling. Similarly, the market's response to order flow, as explored in Chapter 4, could be refined to capture more realistic properties, such as exponential decay in market impact or the emergence of a latent order book structure.

The second direction for future work involves continued refinement and exploration of state-action space design and reward function structure for the RL-based market-maker. There remains substantial scope for experimentation with how bid and ask placements are represented in the action space, how the reward function balances profitability versus inventory management, and how the incorporation of risk-adjusted returns may influence agent behaviour. In addition, the robustness of RL-based market-makers could be tested under market stress scenarios. For example, case studies centred around periods like the February 2020 COVID-19 selloff could provide valuable insights into whether these agents are capable of maintaining liquidity when traditional human market-makers withdraw. Completing these lines of inquiry would enable a more thorough analysis of agent convergence behaviour, trade-offs in learning outcomes, and the relative importance of individual state-action features. Finally, if the simulation framework can be scaled more efficiently, a compelling line of research would be to introduce multiple RL-based market-makers with varying designs into the same environment. This would enable head-to-head comparisons of their effectiveness in liquidity provision under identical market conditions, offering a powerful tool for competition analysis.

References

Frédéric Abergel and Aymen Jedidi. A mathematical approach to order book modeling. *International Journal of Theoretical and Applied Finance*, 16(05):1350025, 2013.

Frédéric Abergel, Marouane Anane, Anirban Chakraborti, Aymen Jedidi, and Ioane Muni Toke. Limit Order Books. Cambridge University Press, 2016.

Jacob Abernethy and Satyen Kale. Adaptive market making via online learning. In *Advances in Neural Information Processing Systems*, pages 2058–2066, 2013.

Eric M Aldrich and Kristian López Vargas. Experiments in high-frequency trading: comparing two market institutions. *Experimental Economics*, 23(2):322–352, 2020.

Eric M Aldrich, Joseph Grundfest, and Gregory Laughlin. The flash crash: A new deconstruction. *Available at SSRN 2721922*, 2017.

Saketh Aleti and Bruce Mizrach. Bitcoin spot and futures market microstructure. *Journal of Futures Markets*, 41(2):194–225, 2021.

Aurélien Alfonsi, Antje Fruth, and Alexander Schied. Optimal execution strategies in limit order books with general shape functions. *Quantitative Finance*, 10(2):143–157, 2010.

José Almeida and Tiago Cruz Gonçalves. Cryptocurrency market microstructure: a systematic literature review. *Annals of Operations Research*, 332(1):1035–1068, 2024.

Yakov Amihud and Haim Mendelson. Dealership market: Market-making with inventory. *Journal of Financial Economics*, 8(1):31–53, 1980.

Selim Amrouni, Aymeric Moulin, Jared Vann, Svitlana Vyetrenko, Tucker Balch, and Manuela Veloso. Abides-gym: gym environments for multi-agent discrete event simulation and application to financial markets. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

Martin Angerer, Marius Gramlich, and Michael Hanke. Order book liquidity on crypto exchanges. *Journal of Risk and Financial Management*, 18(3):124, 2025.

Matteo Aquilina, Eric B. Budish, and Peter O'Neill. Quantifying the high-frequency trading "arms race": A simple new methodology and estimates. Working Paper 300, 2020. URL https://hdl.handle.net/10419/262702.

Leo Ardon, Nelson Vadori, Thomas Spooner, Mengda Xu, Jared Vann, and Sumitra Ganesh. Towards a fully rl-based market simulator. *arXiv preprint arXiv:2110.06829*, 2021.

W Brian Arthur, John H Holland, Blake LeBaron, Richard Palmer, and Paul Taylor. Asset pricing under endogenous expectation in an artificial stock market. Technical report, 1996.

Marco Avellaneda and Sasha Stoikov. High-frequency trading in a limit order book. *Quantitative Finance*, 8(3):217–224, 2008.

Leemon C Baird. Advantage updating. Technical report, Technical Report WL-TR-93-1146, Wright-Patterson Air Force Base Ohio: Wright ⋯, 1993.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

Giuseppe Ballocchi, Michel M Dacorogna, Carl M Hopman, Ulrich A Müller, and Richard B Olsen. The intraday multivariate structure of the eurofutures markets. *Journal of Empirical Finance*, 6(5):479–513, 1999.

Aurelio F. Bariviera. The inefficiency of bitcoin revisited: A dynamic approach. *Economics Letters*, 161:1–4, December 2017. ISSN 0165-1765. . URL http://dx.doi.org/10.1016/j.econlet.2017.09.013.

Emilio Barucci, Giancarlo Giuffra Moncayo, and Daniele Marazzina. Market impact and efficiency in cryptoassets markets. *Digital Finance*, 5(3):519–562, 2023.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Jean-Philippe Bouchaud. Agent-based models for market impact and volatility. *Handbook of Computational Economics*, 4:393–436, 2018.

Jean-Philippe Bouchaud, Marc Mézard, Marc Potters, et al. Statistical properties of stock order books: empirical results and models. *Quantitative Finance*, 2(4):251–256, 2002.

Rasa Bruzgė et al. Stylized facts, volatility dynamics and risk measures of cryptocurrencies. *Journal of Business Economics and Management*, 24(3):527–550, 2023.

Eric Budish, Peter Cramton, and John Shim. Implementation details for frequent batch auctions: Slowing down markets to the blink of an eye. *American Economic Review*, 104(5):418–424, 2014.

Eric Budish, Peter Cramton, and John Shim. The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621, 2015.

David Byrd, Maria Hybinette, and Tucker Hybinette Balch. Abides: Towards high-fidelity multi-agent market simulation. In *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pages 11–22, 2020.

John Cartlidge. ExPo: Exchange Portal. https://sourceforge.net/projects/exchangeportal/, 2020. Accessed: 2020-04-23.

John Cartlidge and Dave Cliff. Evidencing the "robot phase transition"' in human-agent experimental financial markets. In *ICAART-2013: 5th International Conference on Agents and Artificial Intelligence*, pages 345–352. Citeseer, 2013.

John Cartlidge, Marco De Luca, Charlotte Szostek, and Dave Cliff. Too fast too furious: faster financial-market trading agents can give less efficient markets. In *ICAART-2012: 4th International Conference on Agents and Artificial Intelligence*, pages 126–135. SciTePress, 2012.

Damien Challet and Robin Stinchcombe. Analyzing and modeling 1+ 1d markets. *Physica A: Statistical Mechanics and its Applications*, 300(1-2):285–299, 2001.

Nicholas Tung Chan and Christian Shelton. An electronic market-maker. *DSpace@MIT*, 2001.

Yu-Lun Chen, Ke Xu, and J. Jimmy Yang. Market impact of the bitcoin etf introduction on bitcoin futures. *International Review of Financial Analysis*, 97:103810, 2025. ISSN 1057-5219. . URL https://www.sciencedirect.com/science/article/pii/S1057521924007427.

Christopher J Cho and Timothy J Norman. Bit by bit: how to realistically simulate a crypto-exchange. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

Christopher J Cho, Timothy J Norman, and Manuel Nunes. Prime: A price-reverting impact model of a cryptocurrency exchange. *arXiv preprint arXiv:2305.07559*, 2023.

Jeffrey Chu, Saralees Nadarajah, and Stephen Chan. Statistical analysis of the exchange rate of bitcoin. *PloS one*, 10(7):e0133678, 2015.

George Church and Dave Cliff. A simulator for studying automated block trading on a coupled dark/lit financial exchange with reputation tracking. In *Proceedings of the*

31st European Modelling and Simulation Symposium (EMSS2019), pages 284–293. DIME University of Genoa, 2019.

Dave Cliff. Minimal-intelligence agents for bargaining behaviors in market-based environments. *Hewlett-Packard Labs Technical Reports*, 1997.

Dave Cliff. Genetic optimization of adaptive trading agents for double-auction markets. In *Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFEr)(Cat. No. 98TH8367)*, pages 252–258. IEEE, 1998.

Dave Cliff. Zip60: Further explorations in the evolutionary design of trader agents and online auction-market mechanisms. *IEEE Transactions on Evolutionary Computation*, 13 (1):3–18, 2009.

Dave Cliff. An open-source limit-order-book exchange for teaching and research. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1853–1860. IEEE, 2018.

Dave Cliff. Parameterised response zero intelligence traders. *Journal of Economic Interaction and Coordination*, 19(3):439–492, 2024.

David Cliff. Bristol stock exchange. https://github.com/davecliff/BristolStockExchange. Accessed: 2020-04-21.

Andrea Coletta, Matteo Prata, Michele Conti, Emanuele Mercanti, Novella Bartolini, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. Towards realistic market simulations: a generative adversarial networks approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

Lin William Cong, Yilei Dong, Yunbo Lu, Qingsong Ruan, and Guojun Wang. Agent-based modeling for daos and defi. *Available at SSRN 5171559*, 2024.

Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223, 2001.

Rama Cont, Sasha Stoikov, and Rishi Talreja. A stochastic model for order book dynamics. *Operations Research*, 58(3):549–563, 2010.

Wei Cui and Anthony Brabazon. An agent-based modeling approach to study price impact. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–8. IEEE, 2012.

Vince Darley, Alexander Outkin, Tony Plate, and Frank Gao. Sixteenths or pennies? observations from a simulation of the nasdaq stock market. In *Proceedings of the IEEE/IAFE/INFORMS 2000 Conference on Computational Intelligence for Financial Engineering (CIFEr)(Cat. No. 00TH8520)*, pages 151–154. IEEE, 2000.

Rajarshi Das, James E Hanson, Jeffrey O Kephart, and Gerald Tesauro. Agent-human interactions in the continuous double auction. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 1169–1178, 2001.

Sanmay Das. The effects of market-making on price dynamics. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems, volume 2*, pages 887–894, 2008.

Werner FM De Bondt and Richard Thaler. Does the stock market overreact? *The Journal of Finance*, 40(3):793–805, 1985.

Werner FM De Bondt and Richard H Thaler. Further evidence on investor overreaction and stock market seasonality. *The Journal of Finance*, 42(3):557–581, 1987.

J Bradford De Long, Andrei Shleifer, Lawrence H Summers, and Robert J Waldmann. Noise trader risk in financial markets. *Journal of Political Economy*, 98(4):703–738, 1990.

Marco De Luca and Dave Cliff. Agent-human interactions in the continuous double auction, redux-using the opex lab-in-a-box to explore zip and gdx. In *International Conference on Agents and Artificial Intelligence*, volume 2, pages 351–358. SCITEPRESS, 2011a.

Marco De Luca and Dave Cliff. Human-agent auction interactions: Adaptive-aggressive agents dominate. In *Twenty-second International Joint Conference on Artificial Intelligence*, 2011b.

Marco De Luca, CS Szostek, John Cartlidge, and Dave Cliff. Studies of interaction between human traders and algorithmic trading systems. *UK Government Office for Science*, 2011.

Brian Dear. *The Friendly Orange Glow: The Untold Story of the Rise of Cyberculture*. Vintage, 2017.

Nicolás Della Penna and Mark D Reid. Bandit market makers. *arXiv preprint arXiv:1112.0076*, 2011.

Matthew Dixon. Sequence classification of the limit order book using recurrent neural networks. *Journal of Computational Science*, 24:277–286, 2018.

Jonathan Donier and Julius Bonart. A million metaorder analysis of market impact on the bitcoin. *Market Microstructure and Liquidity*, 1(02):1550008, 2015.

Jonathan Donier and Jean-Philippe Bouchaud. From walras' auctioneer to continuous time double auctions: A general dynamic theory of supply and demand. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(12):123406, 2016.

Jonathan Donier, Julius Bonart, Iacopo Mastromatteo, and J-P Bouchaud. A fully consistent, minimal model for non-linear market impact. *Quantitative Finance*, 15(7):1109–1121, 2015.

Andrea Donini, Stefano Grassi, and Manuel Vendolini. Market impact: Empirical evidence, theory, and practice. *arXiv preprint*, 2022.

J Doyne Farmer, Laszlo Gillemot, Fabrizio Lillo, Szabolcs Mike, and Anindya Sen. What really causes large price changes? *Quantitative Finance*, 4(4):383–397, 2004.

Andrea Eross, Frank McGroarty, Andrew Urquhart, and Simon Wolfe. The intraday dynamics of bitcoin. *Research in International Business and Finance*, 49:71–81, 2019.

Fan Fang, Carmine Ventre, Michail Basios, Leslie Kanthan, David Martinez-Rego, Fan Wu, and Lingbo Li. Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, 8(1):13, 2022.

J Doyne Farmer and Duncan Foley. The economy needs agent-based modelling. *Nature*, 460(7256):685–686, 2009.

J Doyne Farmer, Paolo Patelli, and Ilija I Zovko. The predictive power of zero intelligence in financial markets. *Proceedings of the National Academy of Sciences*, 102(6):2254–2259, 2005.

Christopher Fink and Thomas Johann. Bitcoin markets. *Available at SSRN 2408396*, 2014.

Thierry Foucault, Ohad Kadan, and Eugene Kandel. Limit order book as a market for liquidity. *The Review of Financial Studies*, 18(4):1171–1217, 2005.

Sumitra Ganesh, Nelson Vadori, Mengda Xu, Hua Zheng, Prashant Reddy, and Manuela Veloso. Reinforcement learning for market making in a multi-agent dealer market. *arXiv preprint arXiv:1911.05892*, 2019.

Mark B Garman. Market microstructure. *Journal of Financial Economics*, 3(3):257–275, 1976.

Bruno Gašperov and Zvonko Kostanjčar. Market making with signals through deep reinforcement learning. *IEEE access*, 9:61611–61622, 2021.

Bruno Gašperov, Stjepan Begušić, Petra Posedel Šimović, and Zvonko Kostanjčar. Reinforcement learning approaches to optimal market making. *Mathematics*, 9(21):2689, 2021.

Jim Gatheral. No-dynamic-arbitrage and market impact. *Quantitative Finance*, 10(7): 749–759, 2010.

Rémi Genet. Deep learning for vwap execution in crypto markets: Beyond the volume curve. *arXiv preprint*, 2025.

Steven Gjerstad. The competitive market paradox. *Journal of Economic Dynamics and Control*, 31(5):1753–1780, 2007.

Steven Gjerstad and John Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22(1):1–29, 1998.

Dhananjay K Gode and Shyam Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.

Ronald L Goettler, Christine A Parlour, and Uday Rajan. Equilibrium in a dynamic limit order market. *The Journal of Finance*, 60(5):2149–2192, 2005.

Olivier Guéant and Iuliia Manziuk. Deep reinforcement learning for market making in corporate bonds: beating the curse of dimensionality. *arXiv preprint arXiv:1910.13205*, 2019.

Olivier Guéant, Charles-Albert Lehalle, and Joaquin Fernandez-Tapia. Dealing with the inventory risk: a solution to the market making problem. *Mathematics and Financial Economics*, 7(4):477–507, 2013.

Hong Guo, Jianwu Lin, and Fanlin Huang. Market making with deep reinforcement learning from limit order books. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2023. .

SeungOh Han. Price clustering on cryptocurrency order books at a us-based exchange. *Journal of Behavioral and Experimental Finance*, 41:100893, 2024.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: combining improvements in deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Thomas Ho and Hans R Stoll. Optimal dealer pricing under transactions and return uncertainty. *Journal of Financial Economics*, 9(1):47–73, 1981.

Burton Hollifield, Robert A Miller, and Patrik Sandås. Empirical analysis of limit order markets. *The Review of Economic Studies*, 71(4):1027–1063, 2004.

Ruihong Huang and Tomas Polak. Lobster: Limit order book reconstruction system. *Available at SSRN 1977207*, 2011.

Narasimhan Jegadeesh and Sheridan Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91, 1993.

Neil Johnson, Guannan Zhao, Eric Hunsader, Hong Qi, Nicholas Johnson, Jing Meng, and Brian Tivnan. Abrupt rise of new machine ecology beyond human response time. *Scientific Reports*, 3:2627, 2013.

Donald B Keim and Ananth Madhavan. The anatomy of the trading process. *Journal of Financial Economics*, 37(3):371–398, 1995.

Alec N Kercheval and Yuan Zhang. Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, 15(8):1315–1329, 2015.

Robert Kissell, Morton Glantz, and Roberto Malamut. *Optimal trading strategies: quantitative approaches for managing market impact and trading risk*. PublicAffairs, 2003.

Vijay Krishna. *Auction theory*. Academic press, 2009.

Albert S Kyle. Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society*, pages 1315–1335, 1985.

Fredrik Larsen. Automatic stock market trading based on technical analysis. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2007.

Arthur le Calvez and Dave Cliff. Deep learning can replicate adaptive traders in a limit-order-book financial market. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1876–1883. IEEE, 2018.

Blake LeBaron. Building the santa fe artificial stock market. *Physica A*, 1:20, 2002.

Ye-Sheen Lim and Denise Gorse. Reinforcement learning for high-frequency market making. In *ESANN*, 2018.

London Stock Exchange. *Turquoise Plato Block Discovery: Trading Service Description*, version 2.28.0 edition, 2024. URL https://www.lseg.com/markets-products-and-services/our-markets/turquoise/documents. Available from the London Stock Exchange Group (LSEG).

Thomas Lux and Michele Marchesi. Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397(6719):498–500, 1999.

Ana López-Martín et al. Efficiency of cryptocurrency markets: New evidence from interval forecasts. *Finance Research Letters*, 38:101472, 2021.

Mahmoud Mahfouz, Angelos Filos, Cyrine Chtourou, Joshua Lockhart, Samuel Assefa, Manuela Veloso, Danilo Mandic, and Tucker Balch. On the importance of opponent modeling in auction markets. *arXiv preprint arXiv:1911.12816*, 2019.

Kirill Mansurov, Alexander Semenov, Dmitry Grigoriev, Andrei Radionov, and Rustam Ibragimov. Cryptocurrency exchange simulation. *Computational Economics*, 64(5): 2585–2603, 2024.

António Miguel Martins. Short-term market impact of crypto firms'bankruptcies on cryptocurrency markets. *Research in International Business and Finance*, 70:102370, 2024. ISSN 0275-5319. . URL https://www.sciencedirect.com/science/article/pii/S0275531924001636.

Frank McGroarty, Ash Booth, Enrico Gerding, and VL Raju Chinthalapati. High frequency trading strategies, market fragility and price spikes: an agent based model perspective. *Annals of Operations Research*, 282(1-2):217–244, 2019.

Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

Anna A Obizhaeva and Jiang Wang. Optimal trading strategy and supply/demand dynamics. *Journal of Financial Markets*, 16(1):1–32, 2013.

Andrew Phillip, Jennifer SK Chan, and Shelton Peiris. A new look at cryptocurrencies. *Economics Letters*, 163:6–9, 2018.

Silviu Predoiu, Gennady Shaikhet, and Steven Shreve. Optimal execution in a general one-sided limit-order book. *SIAM Journal on Financial Mathematics*, 2(1):183–212, 2011.

K Geert Rouwenhorst. International momentum strategies. *The Journal of Finance*, 53(1): 267–284, 1998.

Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

John Rust, John H Miller, and Richard Palmer. Behavior of trading automata in a computerized double auction market. In *The Double Auction Market*, pages 155–198. Routledge, 2018.

Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

Daniel Savidge and Dave Cliff. Simulation studies of automated trading algorithms for financial exchanges operating frequent batch auctions. In *Proceedings of the 35th European Modeling Simulation Symposium*. 2023.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL http://arxiv.org/abs/1707.06347.

Eduard Silantyev. Order flow analysis of cryptocurrency markets. *Digital Finance*, 1(1): 191–218, 2019.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Eric Smith, J Doyne Farmer, L szl Gillemot, Supriya Krishnamurthy, et al. Statistical theory of the continuous double auction. *Quantitative Finance*, 3(6):481–514, 2003.

Vernon L Smith. An experimental study of competitive market behavior. *Journal of Political Economy*, 70(2):111–137, 1962.

Vernon L Smith. Experiments with a decentralized mechanism for public good decisions. *The American Economic Review*, 70(4):584–599, 1980.

Vernon L Smith. Microeconomic systems as an experimental science. *The American Economic Review*, 72(5):923–955, 1982.

Vernon L Smith and Arlington W Williams. Experimental market economics. *Scientific American*, 267(6):116–121, 1992.

Daniel Snashall and Dave Cliff. Adaptive-aggressive traders don't dominate. In *International Conference on Agents and Artificial Intelligence*, pages 246–269. Springer, 2019.

Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. Market making via reinforcement learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 434–442, 2018.

Steve Stotter, John Cartlidge, and Dave Cliff. Exploring assignment-adaptive (asad) trading agents in financial market experiments. In *ICAART-2013: 5th International Conference on Agents and Artificial Intelligence*, pages 77–88, 2013.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Damian Eduardo Taranto, Giacomo Bormetti, Jean-Philippe Bouchaud, Fabrizio Lillo, and Bence Toth. Linear models for the impact of order flow on prices i. propagators: Transient vs. history dependent impact. *arXiv preprint arXiv:1602.02735*, 2016.

Gerald Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Gerald Tesauro and Jonathan L Bredin. Strategic sequential bidding in auctions using dynamic programming. In *Proceedings of the first International Joint Conference on Autonomous Agents and Multiagent Systems: part 2*, pages 591–598, 2002.

Gerald Tesauro and Rajarshi Das. High-performance bidding agents for the continuous double auction. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 206–209, 2001.

Nicolo Torre. Barra market impact model handbook. *BARRA Inc., Berkeley*, 208, 1997.

Andrew Urquhart. The inefficiency of bitcoin. *Economics Letters*, 148:80–82, 2016.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575:350–354, 2019.

Svitlana Vyetrenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Dervovic, Manuela Veloso, and Tucker Balch. Get real: realism metrics for robust limit order book market simulations. ICAIF '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450375849. . URL https://doi.org/10.1145/3383455.3422561.

Perukrishnen Vytelingum, Dave Cliff, and Nicholas R Jennings. Strategic bidding in continuous double auctions. *Artificial Intelligence*, 172(14):1700–1729, 2008.

Elaine Wah, Mason Wright, and Michael P Wellman. Welfare effects of market making in continuous double auctions. *Journal of Artificial Intelligence Research*, 59:613–650, 2017.

Leon Walras. *Elements of Pure Economics: Or the Theory of Social Wealth*. Routledge, 2013.

Xintong Wang and Michael Paul Wellman. Spoofing the limit order book: An agent-based model. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Ziyi Wang, Carmine Ventre, and Maria Polukarov. Arl-based multi-action market making with hawkes processes and variable volatility. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, page 437 − 444, 2024. ISBN 9798400710810. . URL https://doi.org/10.1145/3677052.3698695.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

Michael P. Wellman, Peter R. Wurman, Kevin O'Malley, Roshan Bangera, Daniel Reeves, William E Walsh, et al. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, 2001.

Michael P Wellman, Amy Greenwald, Peter Stone, and Peter R Wurman. The 2001 trading agent competition. *Electronic Markets*, 13(1):4–12, 2003.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Wei Zhang, Pengfei Wang, Xiao Li, and Dehua Shen. Some stylized facts of the cryptocurrency market. *Applied Economics*, 50(55):5950–5965, 2018.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.

Muchen Zhao and Vadim Linetsky. High frequency automated market making algorithms with adverse selection risk control via reinforcement learning. In *Proceedings of the second ACM International Conference on AI in Finance*, pages 1–9, 2021.

Yuheng Zheng and Zihan Ding. Reinforcement learning in high-frequency market making. *arXiv preprint arXiv:2407.21025*, 2024.