Deep Learning Based Limit Order Book Modelling and Simulation

By

ZIJIAN SHI

Department of Computer Science
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

SEPTEMBER 2023

Word count: fifty-three thousand

Abstract

This thesis is dedicated to utilising state-of-the-art deep learning techniques to facilitate the modelling of the limit order book (LOB), with an ultimate goal to achieve realistic LOB simulation. In financial markets, LOB is a queuing system used to store and sort unexecuted orders submitted by investors, revealing fine-grained contemporaneous demand and supply relation of an asset. LOB has been widely used by researchers and practitioners to gain insights into market dynamics, reflecting its cornerstone role in financial microstructure studies.

The emphasis of the thesis is to conduct realistic LOB simulation, of which the needs originate from practical needs. First, the accessibility of LOB data is very limited, as financial data providers charge a substantial fee for streaming, storage, processing cost associated with the data; and in some scenarios the exchanges do not publish the LOB data at all. Second, traditional LOB simulation models either suffer from subjectivity or lack-of-interaction problems, making the simulated data a weak reflection of the real market. A realistic LOB simulation to be attempted in the thesis can appropriately handle both dilemmas.

Based on a comprehensive overview of the LOB literature, the thesis provides a series of solutions to the aforementioned problems. First, the LOB recreation model is proposed to predict deep level LOB information based on top level trades and quotes data, in which a continuous-time recurrent neural network functions as the main module. While the model allows insights into unrevealed deep level information in the LOB through widely available trades and quotes data, the model fails to simulate a realistic LOB owing to the overlooking of the system's event-based essence. Second, a state-dependent parallel neural Hawkes process is proposed to model event arrivals in financial exchanges as stochastic point processes. The model is able to provide accurate predictions concerning the arrival of next event. Facilitated by stochastic point sampling algorithms, the model can be further utilised to iteratively sample new events, achieving a minimal simulation of the LOB. Finally, the thesis proposes a brand-new perspective in conducting LOB simulation - a hybrid neural stochastic agent-based simulation. The hybrid simulation paradigm embeds a background trader, whose behaviour logic is learnt from real market data using neural point processes and deep generative models, in a well-developed agent-based LOB simulation platform. The hybrid simulation model shares the advantages of both stochastic and agent-based models, leading to an objective and interactive simulation.

Experiments are rigorously carried out to validate the superiority of the proposed models against existing models from the literature. The effectiveness of each proposed deep learning modules, and factors that can potentially influence the performance of the models, are comprehensively discussed. The authenticity of the simulated LOB, and the simulation system's reaction to experimental agents, are verified against a wide range of empirical findings from the financial literature. In conclusion, this thesis underscores the transformative potential of deep learning in enhancing our understanding of financial markets, paving the way for more accurate, efficient, and realistic LOB simulations.

Dedication and acknowledgements

The whole Ph.D. experience has been enormously important and valuable for me. I am grateful to my earlier self who made the decision to take on the challenge, which seemed impossible to conquer at the starting point. I want to express my sincerest gratitude to the people who have made this impossible journey possible. First of all, I want to thank my parents, who have raised me and devoted everything to my development. The sacrifice you made is something that I can never ever pay back enough. Among all others, I want to thank my supervisors, Dr. John Cartlidge and Prof. Dave Cliff. I could never complete my Ph.D. without your support, both academically and mentally. You have always acted as exemplary roles for me, and the rigorous attitude and critical thinking you presented in academic research have inspired me a lot. The eagerness to reach your standard and satisfaction has been the main drive that pushes me forward. Even though my Ph.D. life is ending soon, I will always be your student with pride and honor, and I wish I could be someone like you who devotes and inspires in the days to come. In the end, I want to thank my colleagues and friends who have been working and fighting alongside me throughout the whole process. You people have always been my last resort when I was in needs of help and I am pretty sure I will miss the days we spent together. I want to extend my deepest gratitude to: Dr. Hongbo Bo, Dr. Xinyu Yang, Dr. Yu Chen, Dr. Hugo Herrera, Tian Yang, Jian Ma, Xin Tian, Hanyuan Wang.

List of Publications

The work described in this thesis has been presented in the following publications:

- Zijian Shi, Yu Chen, and John Cartlidge. "The LOB Recreation Model: Predicting the Limit Order Book from TAQ History Using an Ordinary Differential Equation Recurrent Neural Network." AAAI Conference on Artificial Intelligence. AAAI Press, 2020. (covered in Chapter 4, page 45 - page 60)
- Zijian Shi, and John Cartlidge. "The Limit Order Book Recreation Model (LOBRM): An Extended Analysis." ECML-PKDD 2021, European Conference on Machine Learning and Knowledge Discovery in Databases, 2021. (covered in Chapter 4, page 45 - page 60)
- Zijian Shi, and John Cartlidge. "State Dependent Parallel Neural Hawkes Process for Limit Order Book Event Stream Prediction and Simulation." Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022. (covered in Chapter 5, page 61 - page 75)
- Zijian Shi, and John Cartlidge. "Neural Stochastic Agent-Based Limit Order Book Simulation: A Hybrid Methodology." AAMAS 2023, the 22nd International Conference on Autonomous Agents and Multiagent Systems. 2023. (covered in Chapter 6, page 77 - page 111)
- Zijian Shi and John Cartlidge, "Neural Stochastic Agent-Based Limit Order Book Simulation with Neural Point Process and Diffusion Probabilistic Model." Under Review in Intelligent Systems in Accounting, Finance and Management. (covered in Chapter 6, page 77 - page 111)

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Over the past few decades, the landscape of global financial markets has undergone a profound transformation, transitioning from manual operations to sophisticated electronic platforms. Since the first introduction of electronic trading in *New York Stock Exchange* back in 1970s, the shift has been so significant that today, electronic trading accounts for a significant portion of equity shares traded globally. For instance, in the United States, electronic trading accounted for over 99% of equity shares traded in 2020, a stark contrast to the mere 15% in 2000 [1]. This evolution has rendered traditional methods of order submission, such as phone calls or *ticker tape* transmissions, obsolete. They have been replaced by electronic financial exchanges that offer real-time market interaction and efficient data feed APIs, revolutionising the way market participants operate and interact with the market.

The transformation to electronic trading has not only revolutionised traditional financial markets but has also paved the way for the emergence of new platforms that facilitate the trading of a wide range of commodities. Platforms such as *StockX* and *eBay* have leveraged the power of electronic trading to create global marketplaces that transcend geographical boundaries and operate round the clock. The success of these platforms underscores the transformative power of electronic trading. By digitising and automating the trading process, electronic trading has not only increased the efficiency and accessibility of markets but has also fostered innovation, leading to the creation of new trading platforms and markets. As electronic trading continues to evolve, it is expected to continue driving the transformation of global financial markets and beyond.

While the design and structure of markets can vary across different platforms and asset classes, at the core of these electronic trading systems lies the *continuous double auction* (CDA) mechanism [2]. This trading system allows buyers and sellers to submit their orders continuously, with transactions occurring whenever a buy and sell order overlap. The term *continuous* refers

to the fact that there is no restriction on the minimal time interval between consecutive events submitted to the venue, allowing for a constant flow of trading activity. The term *double* signifies that both buyers and sellers can actively engage in the trading process by submitting and cancelling various types of orders. This dual-sided participation creates a dynamic and competitive marketplace where market participants are constantly searching for the best prices. It provides a transparent and fair trading environment where all market participants have equal access to trade execution. The prices formed in this process reflect the collective valuation of the asset by all market participants, making it a crucial component in the price discovery process.

In the framework of the CDA mechanism, a pivotal component is the electronic *limit order book* (LOB), also known as market depth or Level-II data. The LOB is a dynamic data structure that stores and organises all unexecuted orders for a specific financial asset in the market. It is a critical element of the market microstructure, providing granular information about the supply and demand for a given asset [3]. Each entry in the LOB represents an order submitted by a market participant. These orders are sorted by price and time, with *bid* orders (i.e., orders to buy) and *ask* orders (i.e., orders to sell) arranged on different sides of the book. The highest bid and the lowest ask, known as the *best bid* and *best ask*, form the *bid-ask spread*, which is a key indicator of market liquidity [4]. The LOB is a living entity, constantly changing as market participants submit new orders, cancel existing ones, or execute trades in nanosecond time scale. These actions, collectively referred to as *market event stream* (or order flow), shape the state of the LOB and, ultimately, drive price fluctuations. The LOB, therefore, serves as a real-time snapshot of market sentiment, capturing the interplay of various market forces at any given moment. An exemplary snapshot of the LOB for an asset traded in NASDAQ is shown in Fig. 1.1.

The study of the LOB holds substantial importance from both academic and practical perspectives. From an academic viewpoint, the LOB serves as a microcosmic representation of the market, encapsulating the intricate interplay of various market forces. It provides a granular view of market activity, offering a rich dataset for researchers to delve into the complexities of market microstructure, price formation, and market dynamics. The LOB's dynamic nature, reflecting the collective actions and strategies of market participants, makes it a fascinating subject of study for economists, financial theorists, and data scientists alike. It offers a unique lens through which we can understand the underlying mechanisms that drive financial markets, contributing to the broader discourse on economic theory and financial economics [3].

From a practical standpoint, the LOB is of immense relevance to a wide array of market participants. Traders, brokers, market makers, and regulators all rely on the insights derived from the LOB to inform their decisions and strategies. For traders, particularly those engaged in high-frequency and algorithmic trading, the LOB is a critical tool for predicting short-term price movements and devising trading strategies. Brokers and market makers use the LOB to manage liquidity and execute trades optimally, minimising transaction costs and maximising profits. For regulators and policy-makers, the LOB offers a transparent view of market activity,

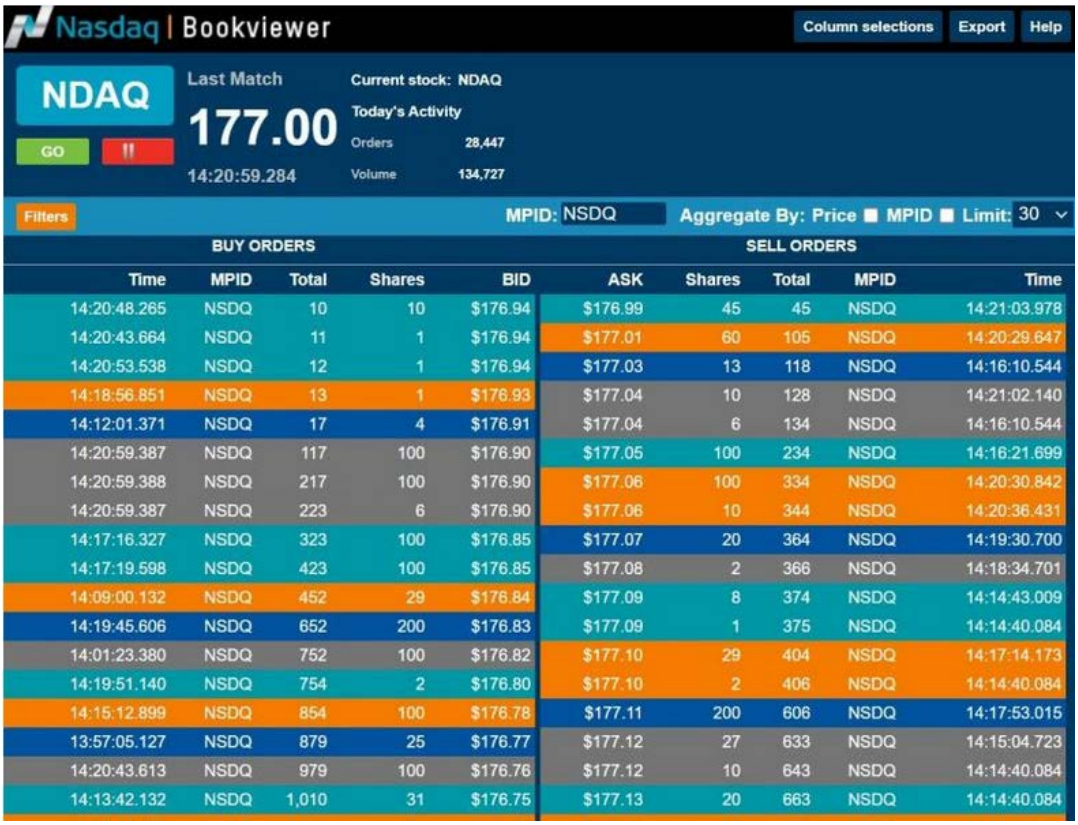


Figure 1.1: An exemplary snapshot of the LOB for the asset *NDAQ* traded in NASDAQ. The price \$177.00 shown on the top indicates the price of the most recent trade. Take the bid side for an example, there are five columns. From left to right, columns indicate time of order submission, ID of the asset, cumulative order volumes with price no lower than the specific price, total number of orders resting at the specific price, and the limit bid price. There are also different batches of orders with the same limit price in the LOB. For instance, at \$176.94 there are three batches of limit orders, submitted by different investors or submitted at different times. The current best bid price is \$176.94 and the best ask price is \$176.99. There are currently 1010 units of orders on the bid side and 663 units of orders on the ask side.

aiding in market surveillance, detecting market manipulation, and ensuring market fairness and transparency [4].

Moreover, the advent of electronic trading and the increasing prevalence of algorithmic trading strategies have further underscored the importance of LOB research. These automated trading systems often rely heavily on LOB data to make trading decisions, making the study of the LOB crucial for their development and performance. As such, research into the LOB not only contributes to our theoretical understanding of financial markets but also has significant implications for the design and operation of trading algorithms and systems [1].

Research into the LOB can be broadly categorised into two main streams: empirical and theoretical. Empirical studies on the LOB have been instrumental in providing a nuanced understanding of the dynamics of financial markets. These studies have delved into the granular

interactions and behaviors that underpin market dynamics, offering a detailed view of market operations. Early empirical studies on the LOB analysed aspects such as order flow, limit order placement, and intraday evolution of the LOB, providing valuable insights into the strategic behavior of traders and the determinants of liquidity in the LOB [5]. Moreover, empirical LOB research has been pivotal in the development and validation of market microstructure theories. Empirical findings have been used to test and refine theoretical models of LOB markets. These studies have helped to bridge the gap between theory and practice, enhancing our understanding of LOB dynamics and informing the design and regulation of financial markets [6, 7].

However, while empirical research provides valuable insights, it is often constrained by the availability and quality of data. High-frequency LOB data is not always readily available, and even when it is, it can be challenging to analyse due to its high dimensionality and complexity. Furthermore, empirical studies are inherently retrospective and descriptive, and while they can uncover patterns and relationships in historical data, they are less suited to predicting future market dynamics or exploring hypothetical scenarios. This is where theoretical research and, more recently, machine learning and AI-based approaches come into play.

Theoretical studies, on the other hand, strive to construct models that can accurately capture and replicate the LOB's dynamics. The task of modelling the LOB is intricate due to the multitude of interactions between market participants and the market that occur on a nanosecond timescale. Each trader, with their unique trading philosophy and strategy, contributes to the temporal evolution of the LOB. The challenge lies in capturing every participant's behavior pattern and building a comprehensive LOB model based on absolute facts, which is a formidable task [3]. In response to this challenge, researchers have sought to strike a balance between model tractability and realism, focusing on essential features that reflect the actual workings of the LOB. Theoretical LOB models can be broadly classified into two categories [3, 8]:

- Agent-Based Models (ABMs): These models use trading agents with pre-defined trading strategies to populate a virtual market, simulating how a real market operates. These models offer a high degree of control, and the relationship between agent parameters and the evolution of the LOB can be explicitly evaluated. They provide a flexible framework for exploring a wide range of market scenarios and trading strategies. Notable examples of ABMs include the work of Cartea et al. [9], McGroarty et al. [10], who developed ABMs to study high-frequency trading strategies; and the one in Foucault et al. [7] developed an equilibrium model to study the strategic behavior of traders in a limit order market, providing valuable insights into the role of limit orders in price formation.

- Stochastic Models (SMs): These models fit the overall order arrival stream to a certain distribution or stochastic point process. They use real market data for parameter estimation, striking a balance between descriptive power and analytical tractability. These models offer a probabilistic perspective on the LOB, capturing the inherent uncertainty and randomness in order arrivals and cancellations. Prominent examples of SMs include the work of Cont

et al. [8], who developed a SM for order book dynamics that has been widely used in the study of LOB, and the work of Cont and De Larrard [11], who developed a Markovian model to study the dynamics of order book markets.

Each of these approaches has its own strengths and weaknesses. They either provide insight into highly stylised model forms learned from real market data or enable market simulation through interactions between virtual agents whose parameters are subjectively set. Despite their limitations, these models have significantly advanced our understanding of the LOB, providing valuable tools for studying market dynamics and informing trading strategies.

While the existing literature has made significant strides in understanding and modelling the LOB, there remains a gap for a model that combines all the desirable features. Specifically, a model that first learns from real market data and then uses this knowledge to produce high-fidelity simulated LOB data [12]. This approach would allow for real-time interactions between the system and experimental agents, enabling the testing of stylised facts that exist in real data or specific trading strategies. This is a compelling feature as most LOB simulators are based on subjective settings of agents or parameters, rather than objectively learned ones from real data. Therefore, such a model has the potential to enrich the current repertoire of LOB simulators used in both research and teaching activities.

Moreover, access to real LOB data often comes at a high cost, with financial exchanges or market data providers charging substantial subscription fees, sometimes amounting to tens of thousands of dollars annually[1]. In some cases, such as in e-commerce platforms or *dark pools* [13], this data may not be available at all. A high-fidelity simulator that produces realistic LOB data could serve as a valuable alternative for backtesting trading algorithms or conducting empirical market studies. This represents a unique application of the simulator compared to traditional LOB simulators, as the learning process is conducted on real LOB data, and the simulated data reflects real market conditions. Despite the potential benefits, few studies have explored this topic.

The rapid advancement of deep learning techniques and high-performance computing in recent years has opened up new avenues for LOB modelling. There has been a growing trend towards applying state-of-the-art deep learning techniques to LOB modelling, with research focusing on developing reinforcement trading agents adaptive to changing market conditions [10], extracting stationary LOB features using recurrent neural networks (RNNs) [14], and generating realistic order streams with generative adversarial networks (GANs) [12].

LOB data, being essentially financial time series data, lends itself naturally to techniques such as RNNs, which have been extensively used in time series modelling tasks in fields as diverse as natural language processing, computer vision, and stock price prediction. The application of deep learning techniques to LOB modelling offers several compelling advantages. Firstly, deep learning models can achieve an improved degree of goodness-of-fit due to the involvement of a

[1] http://www.nasdaqtrader.com/Trader.aspx?id=DPUSdata

large number of artificial neurons in the loss back-propagation process. This allows the models to capture the complex, non-linear dynamics of the LOB with a high degree of accuracy. Secondly, deep learning models are highly flexible and can be adapted to various conditions. With control parameters that can number in the tens of thousands, these models can be fine-tuned to handle a wide range of market scenarios and data characteristics. Thirdly, deep learning models have less stringent assumptions about the data, such as the sampling frequency or order size. This makes them particularly well-suited for dynamic LOB simulations, where the market conditions can change rapidly and unpredictably. In contrast, traditional LOB models often come with strict assumptions and settings, and even slight changes in these can lead to contradictory conclusions. The flexibility and adaptability of deep learning models make them a powerful tool for LOB modelling, capable of capturing the complex dynamics of the LOB and providing valuable insights for trading strategies and market analysis.

To conclude, it is asserted that a research gap exists between the practical needs of empirical LOB studies and the methodological issues associated with theoretical LOB studies. Empirical LOB studies rely on realistic LOB data, which can be either historical or simulated. However, the availability of historical LOB data is limited, necessitating the development of theoretical LOB models to simulate LOB data. An ideal LOB simulation would accurately reflect real market dynamics and be interactive with various experimental settings. Existing literature, however, does not offer an ideal solution for an LOB simulation methodology that combines the advantages of both stochastic models (grounded on real data) and agent-based models (providing an interactive testing environment). Consequently, theoretical LOB models do not meet the needs of empirical studies. In light of this research gap, this thesis aims to contribute to the existing body of theoretical LOB modeling by leveraging deep learning techniques. The goal is to develop a model that can learn from real market data and then use that knowledge to produce high-fidelity simulated LOB data. This simulated data can be interacted with, allowing for empirical studies such as backtesting trading algorithms or investigating causal relations between market variables. Such an approach enables the exploration of hypothetical situations that cannot be tested using historical data alone. This enhanced LOB model allows for the simulation of market dynamics under various scenarios, including different levels of volatility, order flow intensities, or changes in trader behavior through the introduction of interactive agents. This capability empowers researchers and practitioners to test the robustness of trading strategies or market microstructure theories under diverse conditions, thereby deepening their understanding of how the LOB operates in complex real-world scenarios.

The proposed model is expected to serve multiple purposes. For researchers, it can provide a versatile tool for studying market dynamics and testing hypotheses about market behavior. For educators, it can serve as a teaching aid, providing students with a hands-on tool for understanding the LOB and the dynamics of financial markets. For practitioners, particularly those involved in algorithmic trading, the model can serve as a testing ground for developing and backtesting

trading strategies. Moreover, the model can generate realistic LOB data, which can be a valuable resource for researchers and practitioners who do not have access to real LOB data due to cost or availability issues. The simulated data can serve as a proxy for real data, enabling the testing of trading algorithms and the conducting of empirical market studies.

The development of such a model is a challenging task, requiring a deep understanding of both the LOB and deep learning techniques. However, the potential benefits make it a worthwhile endeavor. By combining the power of deep learning with the rich insights offered by the LOB, this thesis aims to advance our understanding of financial markets and contribute to the development of more effective trading strategies and systems. In the following, the motivations, challenges and contributions of the thesis are to be discussed from a high level.

1.1 Research Motivations and Challenges

1.1.1 Modelling the LOB from Trades and Quotes Data

The motivation for modelling the LOB from *trades and quotes* (TAQ) data is multifaceted and extends beyond the practical constraints associated with the procurement and financial implications of LOB data. The limited availability and high cost of LOB data pose significant challenges for researchers and market participants alike. In contrast, TAQ data is readily available and often provided by exchanges at no cost, making it a more accessible source of information for studying market dynamics. The idea of predicting the LOB from TAQ data is not only driven by these practical considerations but also by the potential to gain a more comprehensive understanding of the market. This approach aligns with the broader trend in financial research towards leveraging less traditional data sources to gain novel insights into market dynamics. This could open up new avenues for LOB research and potentially lead to the development of more accurate and efficient LOB prediction and simulation models.

An additional compelling argument for predicting the LOB from TAQ data lies in the potential to gain insights into the hidden fine-grained demand-supply relations in certain trading venues that do not disclose complete LOB information. *Dark pools*, for instance, are private exchanges or forums for trading securities that do not reveal order book information publicly. They are designed to allow participants to trade large quantities of securities without revealing their actions to the public market. The lack of transparency in these venues can make it challenging for investors to understand the underlying dynamics of supply and demand. By predicting the LOB from TAQ data, investors could potentially gain a more nuanced understanding of the hidden demand-supply relations in these venues, thereby informing their trading strategies and decisions.

However, the task of predicting the LOB from TAQ data is not without its challenges. One of the primary challenges is the irregular sampling frequency of TAQ data. Unlike normal time series which is regularly sampled at fixed intervals, TAQ data is event-driven and is sampled

irregularly. This irregularity arises from the fact that trades and quotes occur at random times, driven by the actions of market participants. This irregular sampling frequency poses significant difficulties for traditional modelling techniques, which often assume regular sampling intervals. It necessitates the development of novel methodologies that can handle irregularly sampled data effectively, for which possible candidates can be continuous variants of RNNs that governed by ordinary differential equations [15] or self-decaying kernels [16].

Another significant challenge lies in the high dimensionality of TAQ data. TAQ data is multi-dimensional, encompassing a variety of trade and quote aspects such as price, volume, and time. Moreover, it exhibits multi-modality, with different modes representing different types of events such as new orders, cancellations, and trades. This high dimensionality and multi-modality contribute to the complexity of the data, making it difficult to analyse and model. Traditional modelling techniques often grapple with handling high-dimensional, multi-modal data effectively, resulting in predictions that may not fully capture the intricacies of the data. This necessitates the development of a novel encoding method for TAQ data to enhance the model's comprehension of the input, thereby improving its predictive capabilities.

In conclusion, the motivation for modelling the LOB from TAQ data extends beyond the practical and economic benefits. The potential to uncover hidden demand-supply relations in certain trading venues adds another layer of significance to this approach. Despite the challenges associated with this task, the potential benefits for investors, market transparency, and fairness make it a compelling area of research. The development of effective methods for predicting the LOB from TAQ data could significantly advance our understanding of financial markets and contribute to the development of more equitable and efficient trading systems.

1.1.2 Modelling the LOB from Historical Event Stream

The motivation for modelling the LOB from historical event streams is deeply rooted in the inherent nature of the LOB itself. The LOB is the result from accumulation of continuous market event streams, thereby characterising it as an event-based system [8]. This characteristic is integral to the LOB, and any modelling attempt must duly consider this aspect. However, traditional modelling approaches often overlook this trait, leading to models that may not be suitable for realistic LOB simulation. This significant limitation of existing methodologies underscores the need for alternative methods that can more accurately encapsulate the event-based nature of the LOB.

Empirical studies of the LOB have demonstrated that event streams in financial markets exhibit memory and can be modeled using stochastic processes such as inhomogeneous Poisson processes or Hawkes processes [17, 18]. This provides the possibility of employing a model to learn the patterns inherent in these event streams. Once we have a model that has been trained on market event stream data and has acquired knowledge of its patterns, we can leverage this model to predict forthcoming events. This capability could significantly augment our

ability to comprehend and predict market dynamics, thereby offering a powerful tool for market participants.

Moreover, the utilisation of stochastic sampling methods enables us to stochastically sample event stream data based on the knowledge acquired by the model. This opens up the possibility of conducting LOB simulations, which could serve as a valuable tool for understanding market dynamics and formulating effective trading strategies. The potential to simulate the LOB based on learned patterns of market event streams presents a compelling motivation for modelling the LOB from historical event streams.

However, this approach is not devoid of significant challenges. Traditional stochastic process models often exhibit weak expressiveness, which may hinder the modelling of high-frequency data. High-frequency data is characterised by rapid fluctuations and intricate patterns, which can be challenging to capture with models that have weak expressiveness. This is a significant challenge that necessitates further research to develop effective models for the LOB.

In this context, deep generative models, particularly the neural Hawkes process, have shown promise in tackling these challenges [16]. These models can capture the complex temporal dynamics of high-frequency data and have the expressiveness needed to model the intricate patterns found in such data. The neural Hawkes process, in particular, is capable of learning the intricate temporal patterns of event streams, making it a promising approach for modelling the LOB.

In addition, research has indicated that the arrival of events is influenced not only by historical events but also by other factors such as the current market condition [19, 20]. This complex influence is challenging to model and necessitates an appropriate method that can capture both the influence of historical events and the current market condition. The need to model this complex influence presents a major challenge in the modelling of the LOB from historical event streams.

In conclusion, the motivation for modelling the LOB from historical event streams is driven by the event-based nature of the LOB and the potential to learn patterns of market event streams and simulate the LOB. Despite the significant challenges associated with this approach, including the weak expressiveness of traditional stochastic process models and the need to model complex influences on event arrivals, the potential benefits make it a compelling area of research. The development of effective methods for modelling the LOB from historical event streams could significantly advance our understanding of financial markets and provide valuable alternatives for conducting realistic market simulation.

1.1.3 Combining Neural Stochastic Model with Agent-Based Model - A Hybrid LOB Simulation Towards Realism

The motivation for integrating SMs with ABMs in the context of financial markets is deeply rooted in the inherent limitations of these models when used independently. SMs or deep generative

models have demonstrated their ability to learn the overall market event pattern by grounding the model on real market data. However, these models, in their current form, lack the capability to interact with external strategic trading agents, a crucial aspect of real world financial markets. This limitation restricts their broader application and underscores the necessity for hybrid approaches that amalgamate the strengths of both modelling paradigms.

In this vein, the integration of a model that has learned the overall market event pattern into an agent-based simulation platform presents an intriguing proposition. This would facilitate the observation of interaction dynamics between the market (represented by the neural stochastic model) and strategic traders within an ABM environment. Such a setup could potentially yield valuable insights into how these interactions mirror empirical market findings, thereby enhancing the realism and comprehensiveness of market simulations. This innovative approach could allow us to harness the advantages of both SMs and ABMs, potentially leading to more accurate and insightful market simulations.

However, the path to this integration is fraught with significant challenges. One of the primary challenges lies in enabling the neural stochastic model to generate realistic order streams. The prediction of order type and arrival time, while important, is not sufficient. Other order-related attributes, such as order volume and price, are also crucial. These attributes exhibit complex interdependent distributions and are also influenced by several market conditions. Learning these intricate conditional distributions can be a formidable task. Cutting-edge conditional diffusion models, which have shown promise in modelling complex distributions, could potentially be employed to tackle this problem [21, 22].

Another substantial challenge is the implementation of the neural stochastic agent within an ABM environment, and the subsequent verification of its realism. To address this, we propose the establishment of a comprehensive suite of standalone and interactive experiments. The simulation results can then be scrutinised to ascertain whether they mimic stylised facts or empirical findings of the real LOB. This rigorous testing could provide a robust method for verifying the realism of the hybrid model, thereby enhancing its credibility and applicability.

In conclusion, the motivation for integrating stochastic or deep generative models with ABMs is driven by the desire to overcome the limitations of these models when used in isolation and to harness the potential benefits of a hybrid approach. Despite the significant challenges associated with this approach, including the generation of realistic order streams and the implementation and verification of the neural stochastic agent in an ABM environment, the potential benefits make it a compelling area of research. The development of effective hybrid models could significantly advance our understanding of financial markets, contribute to the development of more effective trading strategies, and potentially revolutionise the field of financial market simulation.

1.2 Research Questions and Contributions

1.2.1 Research Question 1: How to predict LOB information based on historical TAQ time series?

In the realm of financial market microstructure studies, the LOB plays a pivotal role by providing a detailed representation of the demand and supply dynamics for financial assets. However, the LOB data is not always readily available, especially in certain scenarios such as *dark pools*, or its availability might be limited due to cost associated with it. This necessitates the development of alternative methods for analysing the LOB. Addressing this need, this thesis introduces the LOB recreation model (LOBRM), a novel approach to predicting the unrevealed deep level LOB information from TAQ data. This study is characterised by three contributions that not only enhance our understanding of the LOB and its relationship with TAQ data, but also open new avenues for future research in this domain:

- Development of a sparse positional encoding method for TAQ data: This study introduces a novel sparse positional encoding method specifically designed for encoding TAQ data. The utility of this method extends beyond the prediction of LOB information, as it is also proven beneficial for other tasks that necessitate the use of TAQ data, such as the prediction of future stock price movements. The encoding method demonstrates a commendable generalisation ability, thereby facilitating a multitude of tasks.

- Formulation of the LOBRM with continuous RNN backbones: The LOBRM, as formulated in this study, is an intricate composition of an events simulator, a history compiler, and an adaptive weighting scheme, each serving a unique purpose in predicting deep level LOB information from TAQ. The cornerstone of this model is a RNN with continuous latent state controlled by ordinary differential equations or decaying kernels, which exhibits proficiency in modelling irregularly sampled time series. This unique approach enables the model to adeptly manage the complexities and irregularities inherent in financial time series data.

- Extended analysis on factors related to model performance: The study provides an in-depth discussion on the LOBRM, and an exploration into the causes of variation in prediction performance across LOB of different assets. It is found that prediction accuracy is inversely related to the volatility of order volumes resting in the LOB. Furthermore, the study reveals that the influence of stochastic drift on prediction accuracy can be mitigated by increasing the number of historical samples. This extended analysis substantiates the practical applicability of the model in real world scenarios.

1.2.2 Research Question 2: How to predict and simulate market event stream based on historical event stream?

While the LOBRM has made significant strides in predicting the LOB from TAQ data, it overlooks the event-based nature intrinsic to the LOB. This oversight can lead directly to unneglectable drawbacks in the realistic simulation of the LOB. Addressing this gap, this study proposes a state-dependent parallel neural Hawkes process (sd-PNHP) to accurately model the event arrivals in the LOB. This innovative model not only predicts the type and arrival time of the next event but also, when equipped with point sampling algorithms, enables the stochastic sampling of forthcoming event streams. This capability opens up the potential for more realistic simulations of the LOB that are grounded on real data. The contributions of the study can be summarised as follows:

- Advancement of the neural Hawkes process structure: In an effort to enrich the expression of latent state evolution for different types of events, this study refines the structure of the neural Hawkes process. Unlike the original model, which uses a single continuous RNN unit to encode input and time information into the latent state with decaying kernels, this study introduces a parallel stacking structure of multiple units. Each unit governs the evolution of the intensity rate for a different event type of which the parameters are once again shared at discrete input times, thereby enhancing the model's ability to accurately model multi-variate point processes.

- Introduction of an event-state interaction mechanism: This study introduces an event-state interaction mechanism, where the current market state indicator influences the evolution of intensity rates, and reciprocally, the type of each event arrival influences the state transition probability. This mechanism enhances the prediction accuracy for both event type and time, and importantly, it does not hamper the efficient sampling of the event-state stream. This capability is particularly beneficial for stream simulation.

- Realistic market event stream sampling: Upon the successful training of the model, this study demonstrates the application of Ogata's thinning algorithm to stochastically sample event streams based on derived intensity rates. Provided with an initial LOB state, a minimal simulation of the LOB is achieved by accumulating and executing sampled events. By conducting a global sensitivity analysis, some dominant factors that have major impacts on several stochastic properties of the simulated LOB are identified. This minimal while data-validated way provides a brand-new alternative in LOB simulation.

1.2.3 Research Question 3: How to conduct realistic LOB simulation considering both stochastic and agent-based models?

In the realm of financial market simulations, achieving a realistic representation of the LOB is of utmost importance. The emphasis is on creating a robust and dynamic simulation environment that is not only grounded in empirical data but also capable of interacting with external agents. This study proposes an innovative approach that employs a hybrid neural stochastic agent-based model (NS-ABM), integrating advanced machine learning techniques with well-developed agent-based simulation platform to mimic real world market dynamics. The contributions of the study can be summarised as follows:

- Utilisation of the deep diffusion probabilistic model (DDPM) for order attributes distributions learning: Empirical data shows that order-related attributes have significant dependency on various market indicators, such as bid-ask spread and market imbalance. This study leverages the DDPM, a non-parametric deep generative method for learning data distributions, to learn the order-related attributes (such as price and volume) conditioned on various market indicators. The DDPM's learned conditional distributions provide auxiliary information that is essential for LOB simulation, going beyond the order arrival type and time information generated by sd-PNHP.

- Introduction of the neural stochastic background trader (BT) on an ABM platform: The neural stochastic BT, whose actions mimic the aggregation of order events posed by the whole market, is proposed in this study. The BT, equipped with the sd-PNHP and DDPM modules, is incorporated into the open-source ABIDES simulation framework. The BT is shown to present a level of fidelity not previously demonstrated in LOB simulation studies through verifying the existence of a comprehensive list of *stylised facts*. Additionally, the study places a strong emphasis on the interpretability of the results produced by the BT, providing deep insights into the mechanisms that produce these results.

- Investigation on the interactions between the BT and strategic traders: This study introduces a population of trading agents with various *trend* and *value* trading strategies within the simulation. It is demonstrated that the BT reacts realistically to endogenous events caused by other trading agents. The resultant LOB dynamics continue to exhibit the stylised facts of real markets. The simulation is also shown to exhibit order flow impact and financial herding behaviors that are similar to empirical observations of real markets. Lastly, the study presents an explicit investigation into the price impact in the system, discussing how it mimics and deviates from empirical studies. Overall, the study provides valuable insights into conducting realistic LOB simulations.

1.3 Overview of the Thesis

The content of this thesis is arranged into following chapters:

- Chapter 2 presents comprehensive literature review on core topics involved in the thesis, including content on empirical LOB studies, theoretical LOB studies, deep learning models for time series, and the application of deep learning techniques in LOB research.
- Chapter 3 presents the preliminaries involved in the thesis, including LOB preliminaries, continuous RNNs and deep generative models.
- Chapter 4 presents the LOBRM model that predict LOB volume information from TAQ data, a static state-to-state method.
- Chapter 5 presents the sd-PNHP model that predict and simulate market event stream based on historical stream data, a dynamic event-based method.
- Chapter 6 presents the hybrid NS-ABM LOB simulation methodology and a series of verification on the realism of this novel simulation system.
- Chapter 7 presents the conclusions and limitations of the thesis, followed by suggested future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Empirical Research on the LOB

2.1.1 Stochastic Properties of the LOB

The stochastic properties of the LOB have been pivotal in the realm of financial market microstructure research. Over the years, empirical studies have meticulously examined various LOB variables, unveiling intricate patterns and behaviors. Empirical studies of the LOB form the starting point for both researchers and practitioners to understand financial markets, and provides foundations for various assumptions in theoretical LOB modelling. Here, elaborations are made on some of the most important LOB variables: limit orders, market orders, temporal features of the LOB, and characteristics of mid-price return.

Limit Orders To begin with, the role of limit orders and how it shapes the LOB are first examined. Limit orders are specific instructions given by traders to buy or sell a security at a predetermined price or better. They play a crucial role in the formation of the LOB, as they essentially set the parameters for potential trades, waiting for a counterparty to match the specified price. These orders provide depth to the market, allowing participants to visualise supply and demand at different price levels. One of the first systematic empirical LOB studies was conducted by Biais et al. [5] on the *Paris Bourse*. The study indicated that in a LOB market, traders compete with each other to supply and consume liquidity with time and price priority considerations. When the book is thin (i.e., when there are relatively low volume of orders at each price level) or the bid-ask spread is large, the conditional probability that traders submit limit orders instead of market orders is higher in order to lower price cost, and vice versa. Overall arrival rate of limit orders was found to be highest at price levels nearest to quote prices,

an observation that also applies to order cancellation rate [23]. The study also found that the resulting accumulated LOB volumes tend to monotonically increase from quote prices to deeper price levels symmetrically on both the bid and the ask side of the book. Even though sometimes owing to differences in market selections or assumptions, contradictory conclusions were made. For instance, Smith et al. [24] described price distribution of limit orders as uniform, while Bouchaud et al. [25] found it to be power-law distributed around current quote prices. Another example would be that, in Potters and Bouchaud [23] order volumes for *Standard and Poor's Depositary Receipts* shares tended to monotonically decrease from the quote prices, compared with the monotonic increasing conclusion mentioned earlier.

As for the order size of limit orders, it was widely found to appear in round number amounts in units of 100 [26, 27], and in general it follows a power law or log-normal distribution [27, 28]. That is, the possibility of order size is the highest at the minimum quantity and the distribution exhibits a long tail. Further, Bouchaud et al. [25] and Maslov and Mills [28] found that the order size is dependent on the relative price of the order, and the mean size of orders tends to decrease as relative price enlarges.

Market Orders Market orders, in contrast to limit orders, are instructions to buy or sell a security immediately at the best available price. While limit orders contribute to the formation of the LOB, market orders are responsible for its consumption. Foucault et al. [7] described the market order as an order type that expresses high eagerness for order execution compared with limit orders. Glosten [29] were among the pioneers to analyse the role of market orders in determining the bid-ask spread. The author posited that the spread compensates market makers for the risk of trading with an informed trader. This is because market orders, especially large ones, can be indicative of private information. When market participants suspect the presence of informed traders, they adjust their limit orders, leading to a wider spread. Another intriguing aspect of market orders is their correlation with market events. Studies such as Lee [30] have shown that the frequency and size of market orders increase significantly during periods of high volatility or when important financial news is announced. This is because traders seek to adjust their positions quickly in response to new information.

Temporal Features In terms of temporal features of the order flow in the LOB, it was widely accepted that the order flow in the LOB presents certain degree of memory. Sandås [31] studied order flow data for stocks traded in *Stockholm Stock Exchange* and found that the upcoming order flow is dependent on the current status of the LOB and historical order flow. Ellul et al. [32] utilised a multi-nomial logit model on order flow data from *New York Stock Exchange* to investigate the conditional order choice problem. They found that the activity and inactivity of order flows were tend to be clustered, and the bid-ask spread signifies as a strong indicator of the intensity of limit orders. When the spread is large, the LOB tends to elicits limit orders; while when the spread is small, marketable orders arrive more frequently. Moreover, the arrival

rate of bid (or ask) orders tends to increase after upward (or downward) mid-price movements, which indicates momentum in the order flow. By exploiting order flow data from *Australian Stock Exchange*, Hall and Hautsch [33] identified several indicators that play vital roles in influencing upcoming order flow, including but not limited to market depth, queued volume, bid-ask spread and recent mid-price volatility. A similar order clustering phenomenon as in Ellul et al. [32] was also observed. Different from previous opinions, Cao et al. [34] argued that the arrival rates of orders is only of first-order dependency, that is the dependency is only on the current status of the LOB instead of its historical status. Nevertheless, majority of studies pointed us to the conclusion that the arrival of orders tend to cluster in the order flow, instead of being random walk arrivals. Such prolonged memory or clustering in the order flow leads to further explorations on the temporal features in other LOB variable. For instance, multiple empirical studies confirmed that autocorrelation exists in order sign series by assigning +1 to bid orders and -1 to ask orders across different markets [35, 36]. Volatility clustering, referring to the phenomenon where large changes in asset prices tend to be followed by large changes, and small changes tend to be followed by small changes, is also a sub-product of the memory existing in the order flow [37]. This memory effect, enduring over prolonged intervals, suggests that trading strategies were interlinked and shaped by recent market movements.

Mid-Price Return Mid-price, calculated as the mean of the best quote prices, is commonly used in financial markets as a reference point to gauge market direction and is considered a more stable indicator than the last traded price, especially in illiquid or volatile markets. The return calculated on the mid-price time series is used to assess the performance of an asset or to analyse market movements. A seminal work by Fama [38] proposed the normal distribution fit for asset returns, which has been widely used in financial econometrics due to its mathematical tractability. However, following studies challenged this assumption, arguing that asset returns exhibit heavy tails or kurtosis which is not captured by the normal distribution. This observation has been supported by numerous empirical studies conducted on high frequency data from *New York Stock Exchange* [39], *London Stock Exchange* [40] and *Shenzhen Stock Exchange* [41]. Some other studies, including Cont [42] and Gopikrishnan et al. [43], proposed the concept of aggregational normality, indicating that as the time scale over which return is calculated increases, the return distribution moves towards a normal distribution and the heavy tails gradually disappear. The assumption of aggregational normality can be a useful approximation in many contexts, particularly for risk management and portfolio optimisation purposes. While the distribution of returns may exhibit heavy tails at lower levels of aggregation, the tendency towards normality at higher levels of aggregation can simplify analyses and make mathematical models more tractable.

Another important insight into return series is that, unlike that order flow that exhibits long-term memory, return series was found to be of little history dependency. By investigating the autocorrelation function of return series for assets traded in various exchanges, the autocor-

relation coefficient was found to be not significant [44], except for a small negative coefficient on very short time scales [45]. This observation indirectly reflects the efficient market hypothesis [46] indicating that the price in a financial market intrinsically follows a random walk, and all the price changes are caused by the inflow of new information instead of historical information.

2.1.2 Price Impact of Orders

Price in a LOB market is driven by interactions of orders, and the changes in quote prices resulted from trader's actions are termed the price impact [10]. Individual trades happening in the market are consistently interpreted by the market as signals that have both temporary and long-term influence on price [26, 47]. From a high level, price impact consists of two components [48]: instantaneous impact, which is the immediate effect of a specific action, and permanent impact, which is the long-term effect due to a specific action causing other traders to behave differently in the future. Price impact sometimes has negative influence on traders and raise the indirect transaction cost. For example, imagine a trader intends to buy large volumes of an asset. The trader can either put an aggregated large buy market order, or alternatively, divide the large order into several small orders and send them into the market one at a time with delays. The first alternative is more likely to cause a price fluctuation and incur a higher transaction cost, partially owing to rapid consumption of the liquidity at best prices and partially owing to the informed trading behaviours of informed traders.

Instantaneous price impact has been studied extensively in the literature. For instance, Hasbrouck [49] found that the instantaneous price impact of a single market order was larger than the sum of the instantaneous price impacts of two smaller market orders. In order to quantify such impact, researchers have been trying to find a instantaneous price impact function. Most studies agreed on the conclusion that the price impact follows a concave, positive relation function of the individual order volume. That is, the price impact is more sensitive to change in order volume when volume is small, or 'marginal information content of trades is decreasing with order size' [5]. Meanwhile there were disputes over the exact function form of price impact. Empirical findings in *Paris Bourse* indicated a weak concave function that does not depart far from linearity [5], while the relation was found to follow power law in NYSE [50], with a function form of $\Delta p = \frac{\eta v^{\beta}}{\lambda}$ in which Δp is the change in mid-price, η takes value in $+1$ and -1 indication trade direction, v is the trade volume, λ is an adjustment term for market capitalisation and β is the learned parameter. Concerning the parameter of λ, its value varies in existing empirical studies from 0.2 to 0.5 [50–52] due to market differences. There were also debates over the power law fit of price impact, as Potters and Bouchaud [23] and Farmer and Lillo [53] suggested that a logarithmic functional form provided a better fit.

On the other hand, permanent price impact is more difficult to quantify as it involves comparing price difference between scenarios in which the action did occur and those in which it did not occur, while such scenario analysis is impossible to implement in empirical data. As such,

permanent price impact was most commonly measured in the form of long-term price impact, which is to investigate the changes in quote prices and trade imbalance during a certain time period. Kempf and Korn [54] found that the mean mid-price logarithmic return in a 5-minute window for *German Stock Index Futures* followed a concave function of the trade imbalance count during that interval, of which the result was also verified in Gabaix et al. [55]. Based on foreign exchange market, Evans and Lyons [56] reported a significant, positive linear relationship between the daily trade imbalance count and the logarithmic return over successive trading days. Alternatively, Cont et al. [57] proposed to use order flow imbalance instead of trade balance to evaluate long term price impact in the hope of a more robust fit.

In addition to these empirical studies, the literature also discussed the strategies that traders could use to manage possible market impact. For example, traders often split their buy orders to make it more attractive for other participants to submit new sell limit orders [48]. The option consumes liquidity in the market gradually, allowing new sellers time to refill liquidity around the original quote prices. This will result in an overall transaction price around the original, and quote prices will not shift drastically. Another strategy is to place iceberg orders, which is an effective way to hide the true size of limit orders from the market and thus to minimise market impact that might be caused by informed traders.

2.1.3 High Frequency Trading

High-frequency trading (HFT) is a method of trading that uses powerful computers to transact a large number of orders at extremely high speed. More formally, *Securities and Exchange Commission* defined it as 'professional traders acting in a proprietary capacity that engage in strategies that generate a large number of trades on a daily basis' [58]. These high-frequency traders use proprietary algorithms to analyse multiple markets and execute orders based on market conditions. Over the past decade, HFT has become a significant force in the global financial markets, accounting for a substantial portion of the trading volume in many markets. As of 2018, HFT has already taken 52% of all the equity trading volume in the United States [59]. Nowadays, HFT plays a vital role in influencing LOB dynamics.

The role of HFT in the price discovery process has been a key area of investigation. Brogaard et al. [60] found that HFT generally contributes positively to the price discovery process. Their observations indicated that high-frequency traders often trade in alignment with long-term price changes and against temporary pricing errors. In a similar vein, the study by Conrad et al. [61] on U.S. equity markets suggested that high-frequency quoting can lead to price series that more closely follow random walks, an indication of increased efficiency in the price discovery process.

However, it was widely found that HFT leads to increased market volatility. Froot et al. [62] suggested that short-term traders, including high-frequency traders, might exhibit herding behavior when reacting to a single source of information. This could potentially lead to price dislocation and increased volatility. In a similar context, Jarrow and Protter [63] proposed that

the collective reaction of high-frequency traders to common signals could create excessive price pressure and volatility, mimicking the action of a large trader.

Lastly, the influence of HFT on the overall integrity and functionality of the market has been a focal point of discussion and analysis. Carrion [64] emphasised that the comprehension of HFT's ramifications on market quality is still in its infancy. This is primarily attributed to HFT's relatively nascent emergence in the financial landscape and the scarcity of comprehensive and reliable data on the subject. On one hand, several research endeavors have posited that the involvement of HFT in financial markets can yield beneficial outcomes, enhancing liquidity and reducing transaction costs. However, on the flip side, there are mounting worries regarding the potential detrimental effects of HFT. Specifically, there are fears about its role in facilitating predatory trading tactics, a sentiment echoed by Manahov et al. [65]. In the contemporary research landscape, there seems to be a growing consensus supporting the notion that HFT could indeed have adverse consequences for the market's health and quality. This perspective is supported by empirical findings presented in studies like those by Cartea et al. [66] and Shkilko and Sokolov [67], which delved into the intricate dynamics of HFT and its potential pitfalls.

2.2 Theoretical Research on the LOB

Despite the fact that empirical studies on the LOB are rich, formulating theoretical models to explain and validate those findings is a challenging task, with an effort to achieve a balance between model tractability and market realism. It serves as an essential tool to explain empirical observations on the market, and understanding the intricate causal relation between market variables. Both market investors and regulators stand to gain from these theoretical models, with the former seeking profitability avenues and the latter aiming to demystify market anomalies. Here, theoretical LOB models are divided twofold, namely agent based models (ABMs) and stochastic models (SMs) as classified in Cont et al. [8].

2.2.1 Agent-Based Models

ABMs, rooted in the bottom-up simulation approach, create a virtual LOB venue populated with heterogeneous trading agents. The dynamics of the LOB emerge from the intricate web of interactions among these agents that each serve a unique purpose. For instance, the role of market maker agents in ABMs is reminiscent of large security broker-dealers in real markets, providing liquidity on both sides of the book. On the other hand, strategic traders in these models mirror the tactics of real world investors who often ride the wave of price trends. The interaction results between agents and the exchange provides a configurable environment in which we can explore LOB dynamics.

ABMs have been utilised as primary venue to investigate the possible price impact of trading behaviours of agents, or even price manipulation behaviours. One of the noteworthy works is

Wang et al. [68], in which they constructed an ABM populated with zero intelligence (ZI) agents and heuristic belief learning (HBL) agents to dissect the role of spoofing in price manipulation. The trading decisions of both ZI and HBL agents were influenced by noisy observations on an exogenous fundamental value. However, the HBL agents had an edge, leveraging additional LOB information to discern price valuations. The study concluded that the LOB-dependent decision-making process of HBL agents made them susceptible to spoofing, rendering the price more prone to manipulation. Similarly, Zare et al. [69] utilised a Poisson random variable to represent the number of market participants, each being either a seller or buyer with a defined probability. The model established a linear relationship between the percentage of sellers and changes in the logarithm of the price. A discovery, termed the *intrinsic bubble*, revealed that standard dealer activities can falsely inflate stock prices. The study suggested that calculating the daily stock price using the median of traded prices, rather than their mean, could substantially reduce this intrinsic bubble effect.

Moreover, ABMs have also been used to replicate and explain some empirical findings on the LOB, namely *stylised facts*. Diving deeper into the studies, Cont [37] employed an ABM to unearth the factors behind volatility clustering in asset returns. Their model was based on agents trading on perceived market volatility. The findings were revelatory, establishing a connection between the frequency of market activity and agent threshold behavior. The study posited that volatility clustering might be a byproduct of investor inertia. Further, McGroarty et al. [10] introduced an ABM simulation to explore the dynamics of algorithmic trading strategies. This simulation incorporated a fully functioning LOB and featured five distinct types of agents, representing various market behaviors and strategies. When compared with equity market depth data from the *Chi-X Exchange*, the model accurately replicated several empirically observed market behaviors, including autocorrelation of price returns and volatility clustering. Notably, the research found that the presence of HFT agents, under certain conditions, can lead to extreme price events in the market. Also, this high realism of the interaction results produced by the ABM model paved the way for conducting realistic LOB simulation.

In terms of exploring the role of HFT in nowadays financial exchanges, ABMs also play a vital role. In Jacob Leal et al. [70], a market environment where traders operate based on different time-driven strategies was simulated. The analysis uncovered that when high-frequency traders are active, the market experiences heightened volatility and is more prone to sudden, sharp price drops known as flash crashes. Interestingly, the rapid order cancellations by these high-frequency traders, while contributing to market instability, also aid in quicker price recoveries after such crashes. Further, Leal and Napoletano [71] investigated the impact of regulatory measures on the dynamics of HFT. By simulating interactions between various trader types, they tested the effectiveness of interventions such as resting time mandates, different types of circuit breakers, fees for order cancellations, and transaction levies. A central insight from their findings is that while some strategies can mitigate abrupt market fluctuations and decrease the likelihood of

sudden price drops, they might inadvertently prolong the recovery period after such events. This highlights the inherent challenge in crafting policies that both stabilise markets and ensure swift recovery from disruptions. Wah and Wellman [72] used an ABM to explore the latency arbitrage behaviours of HFT in fragmented markets. Due to inherent delays in price quote updates, swift arbitrageurs can capitalise and gain profits when discrepancies arise between these markets. The findings indicated that when markets are fragmented and latency arbitrageurs are active, there was a notable decline in overall market benefits and a drop in liquidity. The study suggested that adopting periodic call markets, which operate in discrete time intervals, can counteract these latency arbitrage opportunities, resulting in a more efficient trading environment.

Yet, the world of ABMs is not without its skeptics. The subjective nature of these models, especially concerning the behavioral patterns and parameters of agents, has been a bone of contention. The study by Preis et al. [73] underscored this sentiment, questioning the feasibility of capturing the multifaceted behavior of individuals through subjectively defined trading rules and parameters.

2.2.2 Stochastic Models

Contrasting with ABMs, SMs offer a different lens to view LOB simulations. These models, anchored in stochastic assumptions over the aggregate order flow, estimate control parameters of the system based on empirical data, trying to achieve a balance between model expressiveness and tractability. Here, three main methodologies are illustrated, namely modelling the LOB as Markovian queuing systems, stochastic point processes, or stochastic partial differential equations.

One of the pioneering studies was by Cont et al. [8]. The study first interpreted the dynamics of a LOB using the principles of continuous-time stochastic processes. Drawing inspiration from queuing systems, the authors modelled the LOB as a Markovian queue where limit orders await execution or cancellation. The model was crafted to be both adaptable to HFT data and to mirror real world behaviors of the LOB, all while maintaining analytical simplicity. By employing methods similar to those used in queuing theory, the authors were able to determine various event probabilities, such as potential shifts in mid-price or the chances of order execution at specific bid levels. The model's practicality and accuracy were validated using trading data from the *Tokyo Stock Exchange*. In a following study by Cont and De Larrard [11], the authors utilised a similar methodology while concentrated on the price impact of orders, indicating that in markets with frequent order arrivals, price volatility increases with the ratio of order arrival intensity to market depth.

The research by Large [18] delved into the robustness of the LOB, emphasising their adaptability following substantial trades. It interpreted high-frequency trading data as a series of distinct, sporadic events in continuous time. The Hawkes point process [74], in particular, is characterised by its ability to capture the influence of past events on the intensity of future

occurrences, making it suitable for studying interactions within the LOB. The intensity rates were determined by both a deterministic function and the integration of past events influence. Using data from the *London Stock Exchange*, the researchers applied this model to each trading day as an independent realisation, optimising the likelihood function for each event type. The findings underscored the LOB's occasional sluggishness in recovering post significant trades, but also highlighted its potential for rapid replenishment when conditions are favorable, presenting a fresh lens to understand the intricacies of the LOB. Following a similar methodology, Muni Toke and Yoshida [20] proposed a parametric model grounded on the assumption that the activities of submitting limit orders, market orders, and order cancellations are governed by point processes with intensities that depend on the state. The authors crafted new functional forms for these intensities and devised models for the positioning and cancellations of limit orders. They also introduced a 'priority index' concept to determine which orders get cancelled in the order book. To calibrate the model, they employed likelihood maximisation for parameter estimation. Comprehensive simulations showed that the model's outcomes align well with real world data from the *Paris Stock Exchange* and outperform the conventional Poisson benchmark.

More recently in Cont and Muller [75], the authors presented an innovative approach to understanding the dynamics of the LOB using stochastic partial differential equations (SPDEs). This methodology was designed to encapsulate the entire LOB, with a particular emphasis on the dynamics of the best bid and ask prices. The SPDE framework they employed was driven by semi-martingales, offering a robust representation of the LOB's stochastic nature. One of the standout features of their model is its ability to be parameterised using a low-dimensional diffusion process, ensuring computational efficiency. Within this framework, the authors delved into a two-factor model, which provided insights into the intertwined behavior of the market price and the depth of the order book. To ensure the model's practical applicability, they calibrated its parameters to mirror statistical features observed in real world order flows. Using high-frequency data from electronic equity markets, the authors validated these predictions, underscoring the model's potential in capturing the nuanced interplay between price dynamics and order flow in modern electronic trading platforms.

2.3 Deep Learning Models

2.3.1 Recurrent Neural Network and Its Variants

Vanilla RNNs Recurrent Neural Networks (RNNs) were developed to address the shortcomings of traditional neural networks in handling sequential data. While conventional networks excel in many tasks, they assume temporal independence among data points, losing their state after processing each point. This treatment makes them ill-suited for data with temporal or spatial relations, such as video frames or words in sentences [76]. RNNs, however, are adept at modelling sequences where elements are interdependent, capturing both time and sequence dependencies

across various scales [77].

In the realm of time dependency modelling, RNNs have distinct advantages over models like Markov chains and hidden Markov models [78]. These traditional models, foundational since the early 20th century, become computationally infeasible with an expanding state space, limiting their use for modelling long-range dependencies [79]. RNNs, however, can efficiently capture these dependencies, with their state representation capacity growing exponentially based on their hidden layer nodes. This capability sets them apart from the computational constraints of Markov models. Even though a RNN's capabilities are limited by its size and architecture, their differentiable nature facilitates gradient-based training [80], enabling the application of optimisation and regularisation techniques.

In a vanilla RNN structure, an RNN cell encodes sequential inputs iteratively into a latent state, in which the previous hidden state is used as a part of the input fed to the next iteration [77]:

$$\mathbf{h}_t = \sigma\left(\mathbf{W}^{hx}\mathbf{x}_t + \mathbf{W}^{hh}\mathbf{h}_{t-1} + \mathbf{b}\right) \tag{2.1}$$

in which $\mathbf{x}_t$ is discrete input of size $x \times 1$, and $\mathbf{h}_t$ is the hidden state of size $h \times 1$ at time t; $\mathbf{W}^{hx}$ and $\mathbf{W}^{hh}$ are matrix-form trainable parameters of size $h \times x$ and $h \times h$, and $\mathbf{b}$ is the trainable bias term of size $h \times 1$; σ is the activation function that introduces non-linearity into the model, of which popular choice can be hyperbolic tangent function in regression or softmax function in classification tasks. A loss function is then used to measure the distance between predictions produced by the model with ground truth, and trainable parameters are optimised using gradient descent algorithm. This formulation empowers the model to capture relations between consecutive inputs of a sequence, enabling it to accumulatively develop an overall understanding of the whole trajectory.

However, vanilla RNNs are not without their challenges. One of the most prominent issues is the vanishing and exploding gradient problem [81]. Due to the recurrent nature of the model, gradients can either diminish to near zero or grow exponentially as they are propagated back through time during training. This leads to difficulties in learning long-term dependencies in sequences, as the model struggles to retain information from earlier timesteps. Another limitation is the computational inefficiency, especially when dealing with long sequences, as the recurrent computation requires sequential processing, making parallelisation challenging [82].

Gated RNNs To address the challenges faced by vanilla RNNs, particularly the vanishing and exploding gradient problem, researchers introduced more sophisticated RNN architectures, notably the long short-term memory (LSTM) and the gated recurrent unit (GRU).

LSTM, introduced in Hochreiter and Schmidhuber [83], was designed to combat the vanishing gradient issue, enabling the model to learn long-term dependencies. The LSTM introduced a memory cell, a unique structure that can maintain its state over time, thereby ensuring that the gradient can be back-propagated without significant diminishment. This memory cell is regulated

by three gates: the input, forget, and output gates. These gates determine how new information is stored, how old information is discarded, and how the value should be output to the next neuron, respectively. Mathematically, the LSTM can be represented as:

$$\mathbf{g}_t = \sigma_{tanh}\left(\mathbf{W}_g^{hx}\mathbf{x}_t + \mathbf{W}_g^{hh}\mathbf{h}_{t-1} + \mathbf{b}_g\right) \tag{2.2}$$

$$\mathbf{i}_t = \sigma_{softmax}\left(\mathbf{W}_i^{hx}\mathbf{x}_t + \mathbf{W}_i^{hh}\mathbf{h}_{t-1} + \mathbf{b}_i\right) \tag{2.3}$$

$$\mathbf{f}_t = \sigma_{softmax}\left(\mathbf{W}_f^{hx}\mathbf{x}_t + \mathbf{W}_f^{hh}\mathbf{h}_{t-1} + \mathbf{b}_f\right) \tag{2.4}$$

$$\mathbf{o}_t = \sigma_{softmax}\left(\mathbf{W}_o^{hx}\mathbf{x}_t + \mathbf{W}_o^{hh}\mathbf{h}_{t-1} + \mathbf{b}_o\right) \tag{2.5}$$

$$\mathbf{s}_t = \mathbf{g}_t \odot \mathbf{i}_t + \mathbf{s}_{t-1} \odot \mathbf{f}_t \tag{2.6}$$

$$\mathbf{h}_t = \sigma_{tanh}(\mathbf{s}_t) \odot \mathbf{o}_t \tag{2.7}$$

Where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are input gate, forget gate, and output gate at time t; $\mathbf{s}_t$ and $\mathbf{h}_t$ are cell state and hidden state of the model at time t; $\odot$ denotes the operation of Hadamard product. The LSTM's ability to regulate its memory has made it a popular choice for many sequence modelling tasks, from machine translation to speech recognition [84, 85].

The GRU, introduced by Cho et al. [86], is a simplified variant of the LSTM. It combined the forget and input gates into a single update gate and merged the cell state and hidden state. This reduction in complexity makes the GRU computationally more efficient than the LSTM, while still retaining the capability to model long-term dependencies. The GRU can be represented as:

$$\mathbf{r}_t = \sigma_{softmax}\left(\mathbf{W}_r^{hx}\mathbf{x}_t + \mathbf{W}_r^{hh}\mathbf{h}_{t-1} + \mathbf{b}_r\right) \tag{2.8}$$

$$\mathbf{z}_t = \sigma_{softmax}\left(\mathbf{W}_z^{hx}\mathbf{x}_t + \mathbf{W}_z^{hh}\mathbf{h}_{t-1} + \mathbf{b}_z\right) \tag{2.9}$$

$$\hat{\mathbf{h}}_t = \sigma_{tanh}\left(\mathbf{W}_n^{hx}\mathbf{x}_t + \mathbf{W}_n^{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_n\right) \tag{2.10}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \hat{\mathbf{h}}_t \tag{2.11}$$

Where $\mathbf{r}_t$ and $\mathbf{z}_t$ are reset gate and update gate at time t respectively; and $\hat{\mathbf{h}}_t$ is the candidate activation vector. GRUs have been shown to perform on par with LSTMs on certain tasks while being more efficient in terms of computation and memory [87].

Continuous-Time RNNs It is worth noticing that, all aforementioned RNN models neglect time intervals between sequential inputs, meanwhile presume that the latent state remains constant during gaps between discrete inputs. Even though vanilla and gated RNNs with discrete updating of the latent state have achieved state-of-the-art performance in tasks like time series modelling [88] and natural language processing [84, 89], they are not a good fit for irregularly sampled time series (i.e., time intervals between sequential inputs are different instead of being constant), in the sense that varied lengths of time intervals may result in different evolution patterns for the latent state.

Compromising ways to tackle this issue are to either concatenate time dimension with feature vectors, or to presume time intervals are uniform, both of which may introduce noise into the model or sacrifice valuable temporal information from original data [90, 91]. A more reasonable means is to model the latent state continuously between sequential inputs to simulate the influence of time on its evolution. Recent studies indicated to decay the latent state between sequential inputs with a pre-defined exponential kernel, during which the latent state tends to be deactivated gradually [16, 90]. In such a RNN-Decay model, the continuous latent state between time t_{i-1} and t_i is defined as:

$$\mathbf{h}_{t_i}{}' = \mathbf{h}_{t_{i-1}} * \exp\left(-f_\theta\left(\mathbf{h}_{t_{i-1}}\right) * (t_i - t_{i-1})\right) \tag{2.12}$$

where f_θ is another neural network that derives the decaying coefficient from the hidden state after receiving last discrete input $\mathbf{x}_{i-1}$, and $\mathbf{h}_{t_i}{}'$ is the latent state just before receiving the next input. Apart from this decaying mechanism during the time intervals between discrete inputs, the updating of hidden state at discrete input times is the same as a vanilla or gated RNN.

As the decaying function for the latent state is pre-defined, the risk of underfitting may arise in such a model. This gave researchers the incentive to device a RNN module in which the latent state dynamics can be learned from the data itself. Neural ordinary differential equation (ODE) was proposed in Chen et al. [15], characterised by a continuous latent state representation of which the evolution process is learned other than pre-defined. This family of neural network models the first-order derivative of the latent state over observed steps instead of its exact state function, generalising discrete updates in neural networks to continuous dynamics. A hidden state $h(t)$ is defined as a solution to an ODE initial value problem:

$$\frac{d\mathbf{h}(t)}{dt} = f_\theta(\mathbf{h}(t), t) \text{ where } \mathbf{h}(t_0) = \mathbf{h}_0 \tag{2.13}$$

in which the function f_θ is a neutral network simulating latent state dynamics with parameter set θ. The gradients of the loss function involving such a network can be derived by the adjoint sensitivity method [92].

In a further work, neural ODE was refined to ODE-RNN to handle the difficulties in modelling irregularly sampled time series [93]. In this specific structure design, neural ODE modules were inserted between sequential inputs, with GRU gating mechanism being used to control the instant transformation of latent state at exact observation time points. The latent state between every two inputs can be derived using an ODE solver as:

$$\mathbf{h}_{t_i}{}' = \text{ODEsolver}(f_\theta, \mathbf{h}_{t_{i-1}}, (t_{i-1}, t_i)) \tag{2.14}$$

The superiorities of the ODE-RNN model over traditional RNN models with discrete updates mainly lie in that, it learns the evolution dynamics of the latent state for any given length of time intervals, and the encoded information can be decoded at any time points as the latent state is defined along continuous time dimension. Experiments of the model on medical record and human movements datasets with irregularly sampled observations all showed enhanced accuracy over discrete RNN schemes [93]. A comparison of different RNN schemes is illustrated in Fig. 2.1.

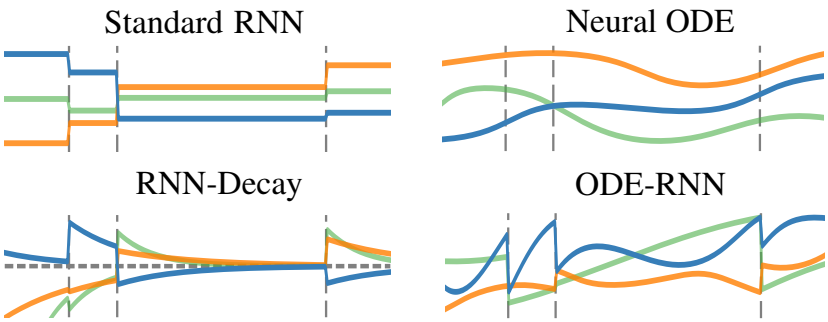


Figure 2.1: Latent state trajectories for different RNN variants. Vertical lines indicate sequential inputs. Different colored lines show different dimensions of the latent state. In standard RNN, latent states between inputs are constant. In RNN-Decay, latent states decay exponentially according to a pre-defined function. Neural ODE learns a continuous latent state along the whole input trajectory. ODE-RNN shares the same terminology with neural ODE, with the only difference being gated updating at exact observation points. Figure is adapted from Rubanova et al. [93].

2.3.2 Deep Generative Models

The digital age has introduced an era abundant with complex data, prompting the creation of advanced models designed to decipher these intricate patterns. Generative modelling is a pivotal area of research in machine learning, with the goal of understanding and emulating the nuanced patterns and structures found in data distributions. Unlike deterministic models, which produce fixed regression or classification predictions without accounting for data uncertainty, generative models learn the entire data distribution. One unique advantage of this modelling methodology is that it allows for stochastic sampling from the learned distribution, generating new data that is from the original distribution while not being exactly the same as prior data. This section delves into three prominent generative models highlighted in the literature.

Variational Autoencoders (VAEs) VAEs emerged from the work of Kingma and Welling [94]. This model marked a significant departure from traditional autoencoders, which were deterministic in nature and primarily used for dimensionality reduction. VAEs introduced a probabilistic approach to generative modelling, aiming to capture the underlying data distribution in a more nuanced manner. The architecture of VAEs consists of an encoder and a decoder. The encoder maps input data into a latent probabilistic space, often represented as a Gaussian distribution. This latent representation is not merely a compression of the data; it captures the salient features, the essence of the data. The decoder then samples from this space, attempting to reconstruct the original data. The objective function of VAEs, known as the evidence lower bound (ELBO), is mathematically represented as:

$$\text{ELBO} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)) \tag{2.15}$$

Where $q(z|x)$ is the approximate posterior, $p(x|z)$ is the likelihood, and $p(z)$ is the prior. This equation ensures a balance between accurate data reconstruction and the preservation of the latent space structure. The KL-divergence term, D_{KL}, acts as a regulariser, ensuring that the latent space does not overfit to the training data. Despite their innovative approach and versatility in various applications, VAEs have faced criticism. One of the most notable critiques is the production of blurry generated samples [95]. This limitation has driven research into refining VAE architectures and loss functions, leading to the development of more advanced VAE variants that address this issue.

Generative Adversarial Networks (GANs) Goodfellow et al. [96] introduced a groundbreaking model known as GAN. GANs represent a radical departure from previous models. Instead of being a single network trying to learn the data distribution, GANs introduced a dual-network architecture. One network, the generator, creates data samples from random noise, while the other, the discriminator, assesses the authenticity of these samples. The objective function of GANs encapsulates an adversarial training mechanism, pushing the generator to produce increasingly realistic samples and the discriminator to become better at distinguishing between real and generated data. This adversarial process is mathematically captured as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2.16}$$

Where $p_{data}(x)$ is the real data distribution and $p_z(z)$ is the noise distribution. This equation represents the competition between the generator and discriminator. GANs have shown remarkable results in various domains, from image generation to style transfer. However, they also have several drawbacks. Training GANs can be unstable, leading to issues like mode collapse [97]. This instability has been a significant research focus, leading to the introduction of various GAN variants. One such variant, the Wasserstein GAN (WGAN) [98], used the Wasserstein distance in its objective, aiming to provide a more stable training landscape.

Deep Diffusion Probabilistic Models (DDPMs) DDPMs offer a unique perspective on generative modelling, drawing inspiration from natural diffusion processes [21, 22]. Instead of trying to learn the data distribution in one go, these models view data generation as a series of transformations. Starting from a basic distribution, the model incrementally adds complexity. At each step in this process, noise is added, and the reverse process involves the systematic removal of this noise, leading to the generation of data samples. This approach provides a granular view of data generation, allowing for a more controlled and nuanced modelling process. However, like all models, DDPMs come with challenges. The most significant of these is the computational demand, especially when simulating the diffusion process. This computational intensity has been a driving force behind ongoing research aiming to develop more efficient training methods and architectures for these models. Fig. 2.2 shows a comparison between the working mechanism of GAN and DDPM.

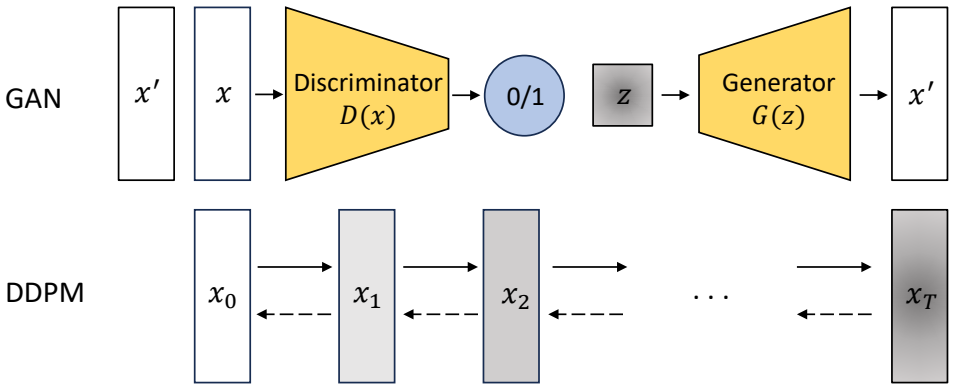


Figure 2.2: Comparison between the GAN and the DDPM. GAN involves a discriminator that tells the difference between genuine samples x and fake samples $x^{'}$, and a generator that generate samples $x^{'}$ from white noise z trying to deceive the discriminator. DDPM tries to learn a bi-directional path between the sample space and the Gaussian space. The diffusion process involves gradually adding Gaussian noise to the samples x (i.e., x_0), and the reverse process involves gradually removing noise from samples from the isotropic Gaussian space.

In conclusion, the journey of deep generative models mirrors the broader trajectory of scientific exploration within machine learning. Nowadays, deep generative models have been widely applied in areas like text-to-image generation [99], medical image reconstruction [100], financial data synthesis [12]. From the early days of deterministic models to the probabilistic foundations of VAEs, the adversarial innovations of GANs, and the nature-inspired mechanisms of DDPMs, the field has seen continuous evolution. Each model, with its unique set of challenges and strengths, represents a step forward in our quest to understand and replicate the world's data.

2.4 Applications of Deep Learning in Modelling the LOB

The benefits of deep learning models are evident across a broad spectrum of applications, attributing to the vast number of parameters in the form of neurons that enhance their reasoning and inference capabilities. In recent years, there has been a surge in research that employs deep learning techniques to leverage and analyse the LOB. This approach seems suitable, especially when traditional models fall short in capturing high-frequency features from the multi-modal data of the LOB. The journey of applying deep learning to the LOB is relatively recent, spanning just the past decade. This thesis categorises the applications of these models into two distinct categories based on their usage, namely LOB prediction and LOB simulation. Other topics like reinforcement learning on the LOB [101, 102] are not discussed here as they are less relevant to the content of the thesis.

LOB Prediction The use of deterministic deep learning models to extract hidden features from LOB series data and then make predictions on various market variables has been the focus

in this research field, of which the most discussed topic is future price movement prediction. One significant study by Sirignano and Cont [103] on a comprehensive pool of 500 stocks revealed that features learned by a LSTM network are stationary and universal for all stocks, and can be used to explain price formation mechanisms. One distinct feature of this research is that it not only paid attention to evaluating prediction accuracy related metrics, but also drew several qualitative conclusions. In particular, their universal LSTM model was able to predict binary next mid-price movement with around 70% accuracy across all 500 stocks, which was overwhelmingly superior than traditional linear models. The authors attributed the performance to the flexibility of using neural networks to represent non-linearity. It was also found that deep learning models suffer less from problems that exist in statistical models of the LOB, such as temporal regime drift, and have improved generalisation ability as discovered universal features are effective for unseen stocks in training. Other qualitative conclusions include that enlarging historical input length can improve prediction accuracy which is indicative of a long memory in price formation dynamics. The success of systematically considering neural networks in modelling the LOB attracted more research to emerge. Ntakaris et al. [104] delved into the challenge of predicting mid-price movements using LOB data. Introducing a novel set of handcrafted econometric features, the authors evaluated their efficacy on various stocks, comparing existing features with those derived from LSTM autoencoders. By employing deep learning models like multi-layer perceptrons, convolutional neural networks, and LSTMs, the research emphasised the potential of these features in accurately forecasting stock price movements. The study also introduced a new experimental protocol for online learning, underscoring the importance of feature selection in high-frequency trading environments. Apart from feature engineering, Zhang et al. [14] combined convolutional filters that capture the spatial structure of LOBs, with LSTM modules that understand time dependencies. Notably, the model outperformed existing state-of-the-art algorithms on the benchmark LOB dataset. When tested using a year's length of market data from the LSE, the model demonstrated consistent out-of-sample prediction accuracy across various instruments. To enhance understanding and transparency, the authors conducted a sensitivity analysis to discern the model's decision-making process, revealing the most influential components of LOBs.

More recent studies have also paid attention to the potential of utilising deep learning models to investigate the execution probability of orders in the market, of which the study is closely related to market making and HFT trading behaviours. Arroyo et al. [105] highlighted the role of determining whether to opt for a passive or aggressive order for trade execution. Central to this decision is understanding the likelihood of a passive limit order being executed within the LOB. The model merged the capabilities of convolutional Transformer encoders with a monotonic neural network decoder. This combination aimed to forecast the duration required for limit orders to be executed across various LOB price levels. The study also offered a comprehensive exploration of the execution probabilities for diverse assets, considering their inherent queue

dynamics and trading patterns. Similarly, Maglaras et al. [106] leveraged the LSTM to capture the temporal dynamics of limit order executions. The study showcased that this deep learning approach can lead to substantial cost savings in a simulated trading scenario. The paper also emphasised the significance of the fill probability in determining optimal trading strategies, especially when considering the trade-offs between market and limit orders.

LOB Simulation However, prior works assumed that full LOB data is available for model training, but unfortunately this is often not the case. Stock exchanges and financial data providers charge a price on the LOB data, which is considerably high especially for individual investors and researchers. Financial market simulators (e.g., *Santa Fe Artificial Stock Market* [107], *Exchange Portal* [108], and *Bristol Stock Exchange* [109]) offer the opportunity to synthesise LOB data using agent-based simulations, but this data may not accurately characterise the real world. More recently, deep learning based LOB simulation methodologies have been investigated to increase the realism in simulated LOBs.

Li et al. [12] introduced the Stock-GAN model, which employed a conditional WGAN to capture the history dependence of orders. The model's design incorporated unique features, such as components approximating the market's auction mechanism that augments order history with LOB constructions to enhance the generation task. They also provided a mathematical characterisation of the distribution learned by the generator and introduced statistics to measure the quality of generated orders. When tested against synthetic and real market data, the Stock-GAN model's generated data closely resembled real data, outperforming several baseline generative models. Prenzel et al. [110] further proposed a GAN based model to incorporate dynamics like temporary drifts, volatility regimes variance, and intraday patterns into the order flow simulation that were commonly neglected by previous models owing to averaging effect during calibration. In Coletta et al. [111], the authors introduced a novel LOB simulation approach that employs a 'world agent' model, trained directly from historical market data. This model aimed to emulate the collective behavior of traders in a LOB market, sidestepping the challenges of calibrating traditional multi-agent simulators. By leveraging a CGAN and a mixture of parametric distributions, the proposed world agent simulator demonstrated superior realism and responsiveness compared to existing methods, offering a more accurate representation of market dynamics.

Based on contemporary literature towards realistic LOB simulation, the best approach would be to first learn the pattern embedded in the real data, and then use such knowledge to produce simulated data. The GAN methodology for realistic order stream generation proposed in Li et al. [12] provides a good example, while the problem lies in that it works like a complete black box as little prior observed stochastic properties of the event stream is considered in the modelling process. For instance, as indicated by empirical LOB studies, event streams in the LOB follow a multi-variate Hawkes process in a general sense [112]. This observation provides potential in using state-of-the-art neural point process models [113], especially the neural Hawkes process [16], to model LOB dynamics. In a neural Hawkes process model, instead of modelling the

individual excitation effect that past events laid on the intensity rate of current event's arrival in a pure stochastic Hawkes process [74], the intensity rate is directly decoded from the hidden state of a continuous-time LSTM unit. In fact, performance of neural Hawkes models have already been tested in a toy LOB event stream dataset, while the experiments were rather only touching the surface [16, 114]. Thus, this thesis intends to deepen the research on combining neural point process and LOB modelling, providing the research community a brand-new perspective in conducting realistic LOB simulations.

CHAPTER 3

PRELIMINARIES

3.1 LOB Fundamentals and the LOBSTER Dataset

In financial markets, trades are governed by a market institution, or namely a mechanism, that specifies certain rules, including legitimate order types, how to match orders into trades and etc. [2]. One dominant market institution that has been adopted by the majority of modern financial exchanges, for example the *New York Stock Exchange* and the *London Stock Exchange*, is the *continuous double auction* (CDA) mechanism. The most characteristic feature of a CDA market is that it allows market participants to enter both buy and sell orders at any time, compared with one-sided auction mechanism that only allow either one of those two types of orders to be continuously submitted (e.g. Dutch auction and English auction). Under CDA mechanism, an electronic LOB is used to record submitted yet unsettled orders in the market. As the LOB provides the most detailed demand and supply information in the market, it is deemed as 'the ultimate microscopic level of description' [25].

A *limit order* $L(t,s,d,v,p)$ specifies a time of submission t, a stock ID s, a direction d, a volume to trade v, and a limit price to trade p, which for a buy order (i.e., a *bid*) is the highest acceptable trade price and for a sell order (i.e., an *ask*) is the lowest acceptable trade price. Each limit order L enters the LOB for a given stock s. The LOB contains a bid list and an ask list, each sorted by price-time priority such that the bid at the front of the bid list (i.e., the *best bid*) has the highest price, $p^{b(1)}$, and the ask at the front of the ask list (i.e., the *best ask*) has the lowest price, $p^{a(1)}$. $p^{b(1)}$ together with $p^{a(1)}$ are termed *quote prices*, and they are deemed as the top price level. The difference between $p^{b(1)}$ and $p^{a(1)}$ is the *bid-ask spread*, and the average of $p^{b(1)}$ and $p^{a(1)}$ is the *mid-price*. $p^{b(n)}$ and $p^{a(n)}$ with $n \in \{2,3,..\}$ are called the deep price levels. When a new bid, b, is entered with price p^b, it will execute against the best ask if $p^b \geq p^{a(1)}$, else b will enter the

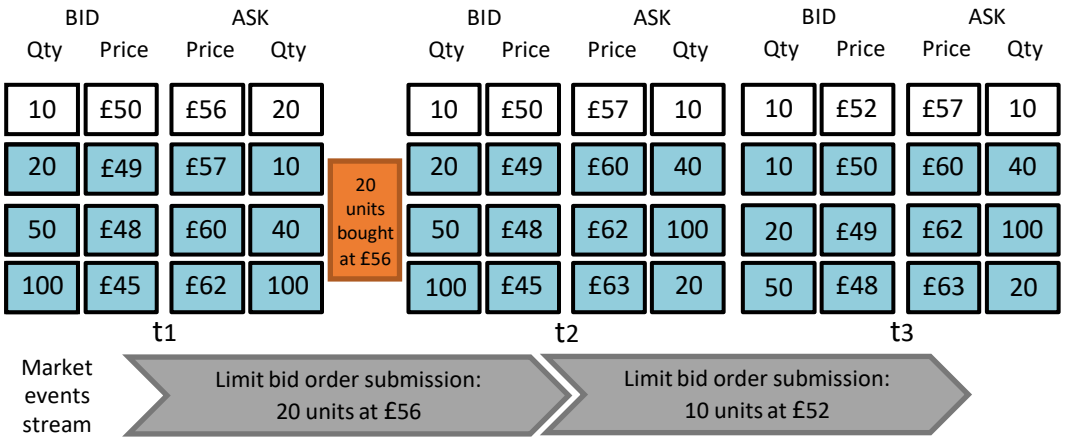


Figure 3.1: A LOB of four price levels evolving with time. White and blue boxes indicate the *top* level and *deep* levels of the LOB. Grey boxes indicate market events stream. Orange box indicates trade records. Three snapshots of the LOB are presented, located at timesteps t_1, t_2 and t_3.

LOB in descending price ordered position. Likewise, when a new ask, a, is entered with price p^a, it will execute against the best bid if $p^a \leq p^{b(1)}$, else a will enter the LOB in ascending price ordered position. Another widely used order type is a *market order*. Such orders usually do not come with a price, and once submitted will be immediately executed against the best available price according to whether it is a bid or ask order. Following this terminology, a market order can also be treated as a limit order with its execution price (normally best bid or ask at trade times) as its affiliated limit price. A LOB is called continuous when there is no limit on time intervals between events, and the term LOB snapshot is used to identify the status of the LOB at a specified timestamp.

Fig. 3.1 presents a schematic of a LOB with four price levels evolving over time. Initially, at time t_1, the best bid has price £50 with volume 10 and the best ask has price £56 with volume 20. The next event at time t_2 is a new bid order submission of price £56 with volume 20. The price of the new bid order is equal to the best ask price, resulting in an immediate transaction of 20 units being bought at price £56. This transaction removes all the orders at the ask quote price and the next ask quote price becomes £57. The following event at time t_3 is a new bid submission with price £52 and volume 10. This order price is higher than the current best bid price £50 while lower than the best ask price. Thus, This order arrives at the top of the bid side of the book and becomes the new best bid.

The full LOB data contains a record of the volume at all price levels in the order book after every event (a trade, or a new order entry or cancellation), along with an affiliated message book that records all market events. In comparison, trades and quotes (TAQ) data, which is generally more freely available than LOB data, includes only the best bid and ask and a record of trades. Although empirical studies have indicated that the first level of the LOB accounts for approximately 80% of future price movements [115], there remains value to exploit in the

information stored deeper in the LOB [5, 116]. This provides motivation for modelling the full LOB from TAQ data, illustrated in Chapter 4.

The dataset used in training and testing for all models in this thesis is from the LOBSTER dataset, provided by the financial data provider LOBSTER [1]. LOB data in the dataset is reconstructed from NASDAQ's historical TotalView-ITCH file, specifically for academic use. The dataset contains LOB data for three *large-tick* stocks, ticker symbol INTC (*Intel*), MSFT (*Microsoft*), and JPM (*J.P. Morgan*), during five trading days from 20/06/2012 to 26/06/2012. Table 3.1 is the sample order book, and Table 3.2 is the affiliated message book, for the stock MSFT. Each line in Table 3.1 is a LOB snapshot resulted from the event update in Table 3.2 of the same line. The LOB data in the dataset is of five price levels (only two price levels are shown here). On average, there are 0.45 million updates in one-day length LOB for each stock (excluding the first half hour after market open, and the last half hour before market close, during which market data is much more noisier).

The LOB data is *continuous* as there is no restrictions on the minimum time interval between market events, resulting in uneven time intervals between LOB updates. The decimal precision for time is nine decimal points (in nanoseconds). There are three types of events in the dataset: 1 indicates order submission; 2 indicates order cancellation; 3 indicates transaction. Each event is associated with an ID to indicate the order being submitted, cancelled or traded. Volume and price are the total volume and limit price for the concerning order. All limit prices are in cents, the minimal tick size. Direction indicates whether the event happens on the bid or the ask side: +1 indicates bid side event and -1 indicates the ask side event.

Table 3.1: A sample of order book for MSFT from the LOBSTER dataset.

time	$p^{a(1)}$	$v^{a(1)}$	$p^{b(1)}$	$v^{b(1)}$	$p^{a(2)}$	$v^{a(2)}$	$p^{b(2)}$	$v^{b(2)}$
36621.79733	307400	5145	307300	18504	307500	19706	307200	15972
36621.79767	307400	5145	307300	19004	307500	19706	307200	15972
36621.79800	307400	5145	307300	17504	307500	19706	307200	15972
36621.79802	307400	5145	307300	17404	307500	19706	307200	15972
36621.79807	307400	4245	307300	17404	307500	19706	307200	15972
36621.79828	307400	4245	307300	17604	307500	19706	307200	15972

Table 3.2: A sample of message book for MSFT from the LOBSTER dataset.

time	type	ID	volume	price	direction
36621.79733	2	48814130	200	307500	-1
36621.79767	1	52087495	500	307300	+1
36621.79800	2	52071243	1500	307300	+1
36621.79802	2	52072553	100	307300	+1
36621.79807	3	48814078	900	307400	-1
36621.79828	1	52087657	200	307300	+1

[1] https://lobsterdata.com/

3.2 Ordinary Differential Equation Recurrent Neural Network

Neural ordinary differential equation (ODE) was proposed in Chen et al. [15] as continuous neural models with continuous hidden states whose derivatives are governed by another neural network. Formerly, the updating of the hidden state in a neural network is achieved by piling a sequence of discrete transformation as:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f_\theta(\mathbf{h}_t) \tag{3.1}$$

Instead, neural ODEs continulised the discrete transformation applied by several neural layers with a derivative of the hidden state of the model as:

$$\frac{d\mathbf{h}(t)}{dt} = f_\theta(\mathbf{h}(t), t) \tag{3.2}$$

where $\mathbf{h}$ denotes the hidden state and f denotes the separate neural network parameterised by $\boldsymbol{\theta}$ that used to derive the derivative of $\mathbf{h}$ concerning time t. By defining the initial value of hidden state $\mathbf{h}(t_0)$, it becomes a initial value problem (IVP) of a dynamic system. By using off-the-shelf ODE solvers like the Euler's method, the final state $\mathbf{h}(t_1)$ can be derived and the same goes to a related loss function $L(\text{ODESolver}(\mathbf{h_0}, \boldsymbol{\theta}, t_0, t_1))$

The main problem for training a neural ODE lies in calculating the gradient of the loss. For a model composed of a series of neural layers, calculation of the gradient requires that all the values of the transitional states be stored, as indicated by the chain rule used in gradient calculation. In the scenario of neural ODEs, storing all the transitional states would be demanding on the hardware which will make the methodology less efficient. Chen et al. [15] proposed to use the adjoint sensitivity method [92]. Such a method takes advantage of an adjoin state $\mathbf{a}(t) = \partial L / \partial \mathbf{h}(t)$ and the gradient can be calculated by solving a second augmented ODE backwards in time. The authors argued that this method scales linearly with time, and is both memory efficient and accurate. Specifically, an augmented adjoint state was defined as:

$$\mathbf{a}_{aug}(t) = \frac{\partial L}{\partial \mathbf{h}_{aug}(t)} \tag{3.3}$$

$$\mathbf{h}_{aug}(t) = \Big[\mathbf{h}(t), \boldsymbol{\theta}\Big] \tag{3.4}$$

where $\mathbf{h}$ is the hidden state and $\boldsymbol{\theta}$ is the trainable parameters. To simplify notation, all vectors are in row representation. The augmented adjoint state follows $\frac{d\mathbf{a}_{aug}(t)}{dt} = -\mathbf{a}_{aug}(t)\frac{\partial \mathbf{f}_{aug}(t)}{\partial \mathbf{h}_{aug}(t)}$.

Proof 1. Let $\frac{d\mathbf{h}_{aug}(t)}{dt} = \mathbf{f}_{aug}(\mathbf{h}_{aug}, \boldsymbol{\theta}, t)$, then $\mathbf{f}_{aug} = \Big[\mathbf{f}, \mathbf{0}\Big]$ as $\boldsymbol{\theta}$ does not change with time. Then:

$$\begin{aligned}
\mathbf{a}_{aug}(t) &= \frac{\partial L}{\partial \mathbf{h}_{aug}(t+\epsilon)} \frac{\partial \mathbf{h}_{aug}(t+\epsilon)}{\partial \mathbf{h}_{aug}(t)} \\
&= \mathbf{a}_{aug}(t+\epsilon) \frac{\partial \mathbf{h}_{aug}(t+\epsilon)}{\partial \mathbf{h}_{aug}(t)} \\
&= \mathbf{a}_{aug}(t+\epsilon) \frac{\partial}{\partial \mathbf{h}_{aug}(t)} \left[\mathbf{h}_{aug}(t) + \int_t^{t+\epsilon} \mathbf{f}_{aug}(\mathbf{h}_{aug}, \boldsymbol{\theta}, t) dt \right] \\
&= \mathbf{a}_{aug}(t+\epsilon) \left[1 + \epsilon \frac{\partial \mathbf{f}_{aug}(\mathbf{h}_{aug}, \boldsymbol{\theta}, t)}{\partial \mathbf{h}_{aug}(t)} \right]
\end{aligned} \tag{3.5}$$

Thus, the derivative of augmented adjoint state of time is denoted as:

$$\begin{aligned}
\frac{d\mathbf{a}_{aug}(t)}{dt} &= \lim_{\epsilon \to 0} \frac{\mathbf{a}_{aug}(t+\epsilon) - \mathbf{a}_{aug}(t)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{\mathbf{a}_{aug}(t+\epsilon) \left[1 - 1 - \epsilon \frac{\partial \mathbf{f}_{aug}(\mathbf{h}_{aug}, \boldsymbol{\theta}, t)}{\partial \mathbf{h}_{aug}(t)} \right]}{\epsilon} \\
&= -\mathbf{a}_{aug}(t) \frac{\partial \mathbf{f}_{aug}(\mathbf{h}_{aug}, \boldsymbol{\theta}, t)}{\partial \mathbf{h}_{aug}(t)}
\end{aligned} \tag{3.6}$$

From Eq.3.6, the mathematical presentation of augmented adjoint state $\mathbf{a}_{aug}(t_0)$ can be obtained by running integration backwards in time from $t_1 (t_1 > t_0)$ as:

$$\begin{aligned}
\mathbf{a}_{aug}(t_0) &= \frac{\partial L}{\partial \mathbf{h}_{aug}(t_0)} \\
&\equiv \left[\frac{\partial L}{\partial \mathbf{h}(t_0)}, \frac{\partial L}{\partial \boldsymbol{\theta}(t_0)} \right] \\
&= \mathbf{a}_{aug}(t_1) + \int_{t_0}^{t_1} \mathbf{a}_{aug}(t) \frac{\partial \mathbf{f}_{aug}(t)}{\partial \mathbf{h}_{aug}(t)} dt
\end{aligned} \tag{3.7}$$

in which the term $\frac{\partial \mathbf{f}_{aug}(t)}{\partial \mathbf{h}_{aug}(t)}$ is a Jacobian matrix of form:

$$\begin{bmatrix} \frac{\partial \mathbf{f}(t)}{\partial \mathbf{h}(t)} & \frac{\partial \mathbf{f}(t)}{\partial \theta} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.8}$$

Essentially, with the help of adjoint sensitivity method, solving for the gradient of loss function L with regards to parameters $\boldsymbol{\theta}$ evolves to solving an IVP of:

$$\frac{d}{dt} \left[\mathbf{h}(t), \frac{\partial L}{\partial \mathbf{h}(t)}, \frac{\partial L}{\partial \boldsymbol{\theta}} \right] = \left[\mathbf{f}(t), -\mathbf{a}(t) \frac{\partial \mathbf{f}(t)}{\partial \mathbf{h}(t)}, -\mathbf{a}(t) \frac{\partial \mathbf{f}(t)}{\partial \boldsymbol{\theta}} \right], \text{ with init. cond. } [\mathbf{h}(t_1), \mathbf{a}(t_1), \mathbf{0}]. \tag{3.9}$$

where $\mathbf{h}(t_1), \mathbf{a}(t_1)$ can be derived through the forward process, and $\boldsymbol{\theta}$ is the initialised parameters. In practice, the system is solved sequentially: first the hidden state, then the adjoint state and finally the loss gradient.

In Rubanova et al. [93], the neural ODE was extended to ordinary differential equation neural network (ODE-RNN) by applying a discrete RNN updating at discrete input timesteps, making it a powerful continuous time series model that is capable of handling irregularly sampled time series. The logic of how a ODE-RNN works is illustrated in Algorithm 1.

Algorithm 1 The ODE-RNN workflow

Input: Data points and their timestamps $\{(\mathbf{x}_i, t_i)\}_{i=1}^{N}$
Initialise hidden state $\mathbf{h}(t_1) = \mathbf{0}$
for $i = 1$ **to** N **do**
    $\mathbf{h}'(t_i) = \text{ODESolver}(\mathbf{h}(t_i), \boldsymbol{\theta}, t_i, t_{i+1})$
    $\mathbf{h}(t_{i+1}) = \text{RNNCell}(\mathbf{h}'(t_i), \mathbf{x}_i)$
end for
$\mathbf{o}_i = \text{Decoder}(\mathbf{h}_i)$ for all $i = 1...N$
Return: $\{\mathbf{o}_i\}_{i=1}^{N}$

3.3 Neural Hawkes Process

A temporal point process is a stochastic process used to describe a series of event arrivals that occur in continuous time. The dynamics of a point process is described through a intensity function $\lambda(t)$. For an infinitesimal time window $(t, t+dt]$, the intensity rate, or the rate of event arrivals, is represented as:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathbb{E}(N(t+\Delta t) - N(t))}{\Delta t} \tag{3.10}$$

in which $N(t)$ is the affiliated counting process that counts total number of arrivals before time t. Various forms of point process have been proposed, to account for an intensity function that is independent of time, dependent on time, or dependent on event history. For instance, a Hawkes process captures the excitation effect of past events on the intensity rate of current event arrivals [74]. The intensity function in a history dependent Hawkes process with exponential kernel is represented as:

$$\lambda(t) = \mu + \sum_{h:t_h<t} \alpha * \exp(-\delta(t - t_h)) \tag{3.11}$$

in which μ, α, δ respectively indicate mean rate, excitation coefficient, and decaying coefficient. Essentially, the equation indicates that in a Hawkes process, the intensity rate at time t is the summation of excitation effect laid by past events, plus an average rate.

Neural point processes [113] differ from a traditional temporal point process in that neural networks are used to govern the evolution of event intensity rate through time. One representative model is the neural Hawkes process [16]. In a neural Hawkes model, rather than modelling the additive individual excitation effect that past events lay on the intensity rate [74], the intensity rate during a time interval is directly decoded from the hidden state of a continuous-time LSTM (CT-LSTM) unit that contains a decaying mechanism. The parameters of the CT-LSTM at i-th discrete inputs updates in the following manner:

$$\tilde{\mathbf{h}}_i^- = \text{concat}((\mathbf{x}_i, \mathbf{h}_i^-) \tag{3.12}$$

$$\mathbf{i}_i = \sigma_{sigmoid}(\text{LC}_{\mathbf{i}}(\tilde{\mathbf{h}}_i^-)) \tag{3.13}$$

$$\bar{\mathbf{i}}_i = \sigma_{sigmoid}(\text{LC}_{\bar{\mathbf{i}}}(\tilde{\mathbf{h}}_i^-)) \tag{3.14}$$

$$\mathbf{f}_i = \sigma_{sigmoid}(\text{LC}_{\mathbf{f}}(\tilde{\mathbf{h}}_i^-)) \tag{3.15}$$

$$\bar{\mathbf{f}}_i = \sigma_{sigmoid}(\text{LC}_{\bar{\mathbf{f}}}(\tilde{\mathbf{h}}_i^-)) \tag{3.16}$$

$$\mathbf{o}_i = \sigma_{sigmoid}(\text{LC}_{\mathbf{o}}(\tilde{\mathbf{h}}_i^-)) \tag{3.17}$$

$$\mathbf{z}_i = \sigma_{tanh}(\text{LC}_{\mathbf{z}}(\tilde{\mathbf{h}}_i^-)) \tag{3.18}$$

$$\mathbf{c}_i = \mathbf{f}_i \odot \mathbf{c}_i^- + \mathbf{i}_i \odot \mathbf{z}_i \tag{3.19}$$

$$\bar{\mathbf{c}}_i = \bar{\mathbf{f}}_i \odot \bar{\mathbf{c}}_{i-1} + \bar{\mathbf{i}}_i \odot \mathbf{z}_i \tag{3.20}$$

$$\boldsymbol{\delta}_i = \sigma_{softplus}(\text{LC}_{\mathbf{d}}(\tilde{\mathbf{h}}_i^-)) \tag{3.21}$$

in which $\mathbf{i}$, $\bar{\mathbf{i}}$, $\mathbf{f}$, $\bar{\mathbf{f}}$, $\mathbf{o}$ are gating controls in a normal LSTM unit; $\mathbf{x}_i$ is the input at timestep t_i after embedding; $\mathbf{c}$ and $\bar{\mathbf{c}}$ are the initial cell state and the cell state decaying target; $\mathbf{h}_i^-$ is the hidden state just before receiving input at time t_i; $\boldsymbol{\delta}$ is the parameter to control decaying pace; LC represents linear connection layer; σ is the non-linear activation function of various forms; $\odot$ is the operation of Hadamard product. During $(t_i, t_{i+1}]$, the cell state, hidden state and intensity rate at time t can be derived as:

$$\mathbf{c}(t) = \bar{\mathbf{c}}_i + (\mathbf{c}_i - \bar{\mathbf{c}}_i)\exp(-\boldsymbol{\delta}_i(t - t_i)) \tag{3.22}$$

$$\mathbf{h}(t) = \mathbf{o}_i \odot \sigma_{tanh}(\mathbf{c}(t)) \tag{3.23}$$

$$\boldsymbol{\lambda}(t) = \sigma_{softplus}(\text{LC}_{\lambda}(\mathbf{h}(t))) \tag{3.24}$$

where the intensity rate $\lambda(t)$ is decoded from the continuous hidden state $\mathbf{h}(t)$ using a Softplus activation function to force its value above zero. According to Daley and Vere-Jones [117], given an intensity function $\boldsymbol{\lambda}(t)$, the conditional density function that the next event happens at time t conditioned on event history $\mathscr{H}_i$ can be derived as:

$$p_i(t) = P(t_i = t \mid \mathscr{H}_i) = \lambda(t)\exp\left(-\int_{t_{i-1}}^{t} \lambda(s)ds\right) \tag{3.25}$$

For the ease of gradient computation, the sum of logarithm of $p(t)$ is used instead in deriving the loss function. The log-likelihood of the model parameters given an event sequence can be computed as:

$$L = \sum_{i:t_i \le T} \log\lambda_{x_i}(t_i) - \int_0^T \lambda(s)ds \tag{3.26}$$

Proof 2. Given complete observations of an event stream over the time interval $[0,T]$, the log-likelihood of the parameters is:

$$L = \sum_{i:t_i \le T} \log \lambda_{x_i}(t_i) - \int_0^T \lambda(s)ds \tag{3.27}$$

where λ_{x_i} is the intensity rate of event type x_i at time t_i, and λ is the sum of intensity rates for all types of events. First, we define $N(t)$ to be the count of events (of any type) preceding time t. Given the past history H_i, the number of events in $(t_{i-1}, t]$ is:

$$\Delta N(t_{i-1}, t) = N(t) - N(t_{i-1}) \tag{3.28}$$

Let $T_i > t_{i-1}$ be the random variable of the next event time. The cumulative distribution function and probability density function of T_i (conditioned on $\mathscr{H}_j$) are:

$$\begin{aligned} F(t) &= P(T_i \le t) \\ &= 1 - P(T_i > t) \\ &= 1 - P(\Delta N(t_{i-1}, t) = 0) \\ &= 1 - \exp\left(-\int_{t_{i-1}}^{t} \lambda(s)ds\right) \\ &= 1 - \exp(\Lambda(t_{i-1}) - \Lambda(t_i)) \end{aligned} \tag{3.29}$$

$$f(t) = F'(t) = \exp(\Lambda(t_{i-1}) - \Lambda(t_i))\lambda(t) \tag{3.30}$$

where $\Lambda(t) = \int_0^t \lambda(s)ds$. Given the past history $\mathscr{H}_i$ and the next event time t_i, the distribution of x_i is:

$$P(x_i = k | t_i) = \frac{\lambda_k(t_i)}{\lambda(t_i)} \tag{3.31}$$

where $k \in \{1, 2, ...\}$ indicating different event types. From this, the likelihood function is derived as:

$$\begin{aligned} \ell &= \prod_{i:t_i \le T} [f(t_i) P(x_i = k | t_i)] \\ &= \prod_{i:t_i \le T} \left[\exp(\Lambda(t_{i-1}) - \Lambda(t_i))\lambda_{x_i}(t_i)\right] \end{aligned} \tag{3.32}$$

Thus, the logarithm likelihood function can be denoted as:

$$\begin{aligned} L &= \log \ell \\ &= \sum_{i:t_i \le T} \left[\log \lambda_{x_i}(t_i) - (\Lambda(t_i) - \Lambda(t_{i-1}))\right] \\ &= \sum_{i:t_i \le T} \log \lambda_{x_i}(t_i) - \Lambda(T) \\ &= \sum_{i:t_i \le T} \log \lambda_{x_i}(t_i) - \int_0^T \lambda(t)dt \end{aligned} \tag{3.33}$$

3.4 Deep Diffusion Probabilistic Model

Formally, a deep diffusion probabilistic model (DDPM) [21] is a latent variable model with $p_\theta(\mathbf{x_0}) := \int p(\mathbf{x}_{0:T})d\mathbf{x}_{1:T}$, in which $\mathbf{x}_1, ..., \mathbf{x}_T$ are latent representations of the same dimensionality as $\mathbf{x}_0$. The joint probability of $p_\theta(\mathbf{x}_{0:T})$ is derived from the reverse diffusion process, during which the transition probability of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is Markovian and subject to learned Gaussian rules parameterised by θ:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{3.34}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \beta_t \mathbf{I}) \tag{3.35}$$

$$p(\mathbf{x}_T) := \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}) \tag{3.36}$$

The transitional representations of $\mathbf{x}_0$ are retrived through a pre-defined Gaussian noise-adding scheme $\boldsymbol{\beta} = \{\beta_t \in (0,1)\}_{t=1}^{T}$ and $\beta_1 < ... < \beta_t < ... < \beta_T$, starting from $\mathbf{x}_0$. The conditional probability of $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ can be denoted as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \tag{3.37}$$

Using reparameterisation and the additivity of Gaussian distributions, it is possible to get the closed form representation of $\mathbf{x}_t$ directly from its initial state $\mathbf{x}_0$. By denoting $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, transitional state $\mathbf{x}_t$ can be rewritten as:

$$\begin{aligned} \mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} \\ &... \\ &= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0 \end{aligned} \tag{3.38}$$

where $\boldsymbol{\epsilon}_0, ..., \boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus, when T is big enough, the distribution of $\mathbf{x}_T$ follows an isotropic Gaussian distribution. Such a process is called the forward diffusion process, during which small amounts of noise are added to original data $\mathbf{x}_0$ until the maximum diffusion step is reached. The forward diffusion process distinguishes from the reverse diffusion process in the sense that the diffusion is fully controlled by the diffusion scheme $\boldsymbol{\beta}$, and the forward diffusion process does not contain any learnable parameters.

Using results in Eq. 3.37 and Eq. 3.38, the probability $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ based on the Bayes theorem during reverse diffusion can be denoted as:

$$
\begin{aligned}
& q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) \\
&= q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&\propto \exp\Big(-\frac{1}{2}\big(\frac{(\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t}+\frac{(\mathbf{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}}-\frac{(\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big(\frac{\mathbf{x}_t^2-2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1}+\alpha_t\mathbf{x}_{t-1}^2}{\beta_t}+\frac{\mathbf{x}_{t-1}^2-2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1}+\bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}}-\frac{(\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2-(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1}+C(\mathbf{x}_t,\mathbf{x}_0)\big)\Big) \\
&\propto \mathcal{N}(\mathbf{x}_{t-1};\frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t),\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathbf{I})
\end{aligned}
\tag{3.39}
$$

i.e., $\mathbf{x}_{t-1}$ given $\mathbf{x}_t$ and $\mathbf{x}_0$ follows a Gaussian distribution with mean and variance shown.

In order to learn the reverse diffusion process, the training target of the DDPM is to maximise the log-likelihood $\log p_\theta(\mathbf{x}_{0:T})$, of which the variational lower bound can be denoted as:

$$
\begin{aligned}
\log p_\theta(\mathbf{x}_0) &\geq \log p_\theta(\mathbf{x}_0) - \boldsymbol{D}_{KL}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)||p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\log\frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}
\end{aligned}
\tag{3.40}
$$

By implementing a series of simplifications shown in Ho et al. [21], the training target downgrades to minimise the Kullback-Leibler divergence between $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. According to Eq. 3.35 and Eq. 3.39, the target is essentially forcing $\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)$ to approximate $\frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t)$. As the value of $\mathbf{x}_t$ is deterministic in the forward diffusion process, the training target can be further simplified to minimise the real and predicted error term:

$$
\begin{aligned}
L &= \mathbb{E}_{t\sim[1,T]}||\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)||^2 \\
&= \mathbb{E}_{t\sim[1,T]}||\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t,t)||^2
\end{aligned}
\tag{3.41}
$$

From the loss function, it can be seen what the neural network essentially does in a DDPM is to make predictions on the error term added from $\mathbf{x}_0$ to $\mathbf{x}_t$ during the forward diffusion process. The predicted error term can then be inserted into Eq.3.39 to derive $\mathbf{x}_{t-1}$.

The DDPM can be trained using normal gradient descent optimisation. After the DDPM converges, the bi-directional mapping between the original data space and isotropic Gaussian space is established. By sampling from $\mathcal{N}(\mathbf{0},\boldsymbol{I})$ and then applying a series of transformations as shown in Eq. 3.35, samples from the original data space can be drawn.

To further utilise the conditional information, classifier-free guidance [22] suggested incorporating the condition into the error prediction term as $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t,\mathbf{c})$, where $\mathbf{c}$ is the conditional information (i.e., market indicators) after embedding, and then replacing the original error term

in Eq. 3.41. During sampling, the final predicted error term is calculated using a weighting scheme w to combine the conditional error and unconditional error:

$$\tilde{\boldsymbol{\epsilon}}_t = (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \tag{3.42}$$

The full training and sampling scheme of the conditional DDPM is illustrated in Algorithm 2 and 3.

Algorithm 2 Training the DDPM with classifier free guidance

Require: p_{uncond}: possibility of unconditional training
1: **repeat**
2: $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$
3: $\mathbf{c} = \mathbf{0}$ with probability p_{uncond}
4: $t \sim \text{Unif}(\{1, ..., T\})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: take gradient step on $\nabla_\theta ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2$
6: **until** convergence

Algorithm 3 Conditional sampling with classifier free guidance

Require: w: guidance strength
Require: $\mathbf{c}$: conditioning information for conditional sampling
Require: $\lambda_1, \ldots, \lambda_T$: noise scheme with $\lambda_1 = \lambda_{\min}$, $\lambda_T = \lambda_{\max}$
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ else $\mathbf{z} = \mathbf{0}$
4: $\tilde{\boldsymbol{\epsilon}}_t = (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$
5: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\tilde{\boldsymbol{\epsilon}}_t) + \sigma_t \mathbf{z}$
6: **end for**
7: **return** $\mathbf{x}_0$

CHAPTER 4

MODELLING THE LOB FROM TAQ: A STATIC VIEW

The content to be presented is a generalisation of previously published works [118, 119].

4.1 Introduction

Financial markets operate through a complex system wherein the LOB plays a crucial role. The LOB is a structured record of buy and sell orders for a specific asset, providing insights into market demand and supply relations and price trajectories. However, comprehensive LOB data is often restricted due to high costs, limiting its accessibility to a broader audience. In contrast, trades and quotes (TAQ) data, which encompasses trade executions and the topmost orders in the LOB, is more readily available. Given the differences in accessibility and cost between LOB and TAQ data, an important research question emerges: Can TAQ data be effectively utilised to reconstruct the detailed structure of the LOB?

Recent research in Blanchet et al. [120] has attempted this question, demonstrating the feasibility of extracting significant information about the LOB using TAQ data. The LOB was depicted as a Markovian multi-class queuing system. In the system, the customers are the limit orders waiting to be traded, and service is provided by market orders, which lead to transactions against limit orders upon their arrivals. Under certain assumptions, each side of the LOB evolves as a set of independent infinite-server queues. Each of these queues corresponds to a price level and consists of the limit orders at that level. The authors used a heuristic approach to approximate the distribution of the queues in the LOB at specific times. Between two given times, the infinite-server queue of limit orders at a relative price has a certain arrival rate and cancellation rate. Consequently, the steady-state distribution of the number of standing limit orders at that relative price is Poisson, with the mean being tractable given empirical data. One

drawback of the research is that it cannot approximate the LOB volume information at a high frequency time scale, making it less capable in high frequency LOB reconstruction.

Building upon the foundational research, this chapter presents a methodology rooted in deep learning techniques to reconstruct the LOB using TAQ data. This approach is termed a 'static view' because it predicts the current snapshot of the LOB from a sequence of past TAQ states, rather than recognising that the LOB continuously updates based on events arrivals. As a result, the simulation of the LOB is realised by sequentially stacking LOB predictions, bypassing the consideration of dynamic market events that lead to LOB updates. The limitations of this static approach are explored at the end of this chapter, which pave the way for subsequent research detailed in this thesis. Nonetheless, the insights shared in this chapter serve as a pivotal connection between LOB and TAQ data, laying the groundwork for LOB simulation.

This chapter presents the LOB recreation model (LOBRM), a deep learning model to predict the LOB from TAQ data. First, in face of the fact that the TAQ data is both multi-dimensional and multi-modal, the one-hot positional encoding method is proposed to encode TAQ data. This sparse encoding method uses relative price information to indicate the position in a one-hot vector where volume information to be filled in, achieving robust performance. The main structure of the LOBRM consists of three modules:

- History Compiler (HC): This module utilises a gated recurrent unit (GRU) to compile prediction-relevant quote history from TAQ data, based on the observation that a significant portion of the LOB persists at sufficiently large time scales. As quote prices are constantly fluctuating, historical quote volume information may well reveal information concerning deep price levels of the current LOB.

- Events Simulator (ES): This component employs a continuous-time recurrent neural network (RNN) to simulate net order arrivals. By representing market dynamics as differential equations, the ordinary differential equation recurrent neural network (ODE-RNN) offers a continuous-time representation of the market, from which the intensity rates of order arrivals are decoded. By integrating the intensity rates over the prediction period, the results are treated as the incremental volume on top of the HC prediction.

- Weighting Scheme (WS): To ensure optimal results, the model integrates an adaptive weighting mechanism that combines the predictions from HC and ES, based on the timeliness and reliability of HC prediction.

In the experiment section, first the performance of the proposed LOBRM model is compared with other mainstream linear and non-linear models. Further, an ablation study is conducted to investigate the contribution of each module to prediction accuracy. Finally, some empirical findings are drawn, concerning (i) the generalisation ability of the proposed encoding method for TAQ; (ii) the cause of variances in prediction accuracy across LOBs of different equities; and (iii) the alleviation of the influence of stochastic drift on the model by enlarging historical data size.

4.2 Problem Description

The core target of the LOBRM is to predict the current LOB information, especially order volumes resting at deep price levels, from TAQ data trajectories. The model is essentially a time series prediction model, constructing a mapping between a defined length of historical TAQ data and a snapshot of LOB.

To simplify the problem of recreating the LOB, following assumptions are made: (i) The model is specifically for *large-tick* stocks. *Large-tick* stocks are highly liquid stocks that commonly have tight bid-ask spread, which makes the *tick size* (i.e., the smallest increment permitted in quoting or trading a security at a particular exchange venue) comparatively large. It is assumed that the price interval between each price level is exactly one tick, which is supported by empirical evidence that orders in the LOB for *large-tick* stocks tend to be densely distributed around the top price levels [121, 122]. Following this assumption, the price at different LOB levels can be directly deduced from the known quote price at the target time. Therefore, the LOB recreation task resolves to the simpler problem of only predicting order volumes resting at each price level; (ii) Following common practice (e.g., Blanchet et al. [120] and Horst and Kreher [123]), the bid and ask sides of the LOB are modelled separately; (iii) The model only considers instantaneous LOB data at the time of each trade event to align with TAQ data, and ignore the LOB at the arrivals of all other events, such as limit order submission and cancellation. Essentially, the LOB attempted here is a resampled LOB at a lower frequency; (iv) The model only considers the top five price levels of the LOB, as information concerning further deeper price levels is deemed less relevant to the evolution of the LOB, a practice also conducted in Cont et al. [8] and Biais et al. [5].

For generalisation, TAQ streams are denoted as $\{TD_i\}_{i=1}^N$ and $\{QT_i\}_{i=1}^N$ respectively, and trajectories of time points for TAQ records as $\{t_i\}_{i=1}^N$ indexed by $i = \{1,\dots,N\}$, where N equals the number of time points in TAQ. The LOB sampled at $\{t_i\}_{i=1}^N$ is denoted as $\{LOB_i\}_{i=1}^N$. For each record at time t_i, $QT_i = \left(p_i^{a(1)}, v_i^{a(1)}, p_i^{b(1)}, v_i^{b(1)}\right)$, where $p_i^{a(1)}, v_i^{a(1)}, p_i^{b(1)}, v_i^{b(1)}$ denote best ask price, order volume at best ask, best bid price, and order volume at best bid, respectively. $TD_i = \left(p_i^{td}, v_i^{td}, d_i^{td}\right)$, where $p_i^{td}, v_i^{td}, d_i^{td}$ denote price, volume, and direction of the trade, with $+1$ and -1 indicating orders being sold or bought. $LOB_i = \left(p_i^{a(l)}, v_i^{a(l)}, p_i^{b(l)}, v_i^{b(l)}\right)$ depicts the price and volume information at all price levels, with $l \in (1,...,L)$, here $L = 5$. From the aforementioned model assumptions, it can deduced that $p_i^{a(l)} = p_i^{a(1)} + (l-1)\tau$ and $p_i^{b(l)} = p_i^{b(1)} - (l-1)\tau$, where τ is the minimum tick size (one cent in the majority of financial exchanges). For a single sample, the model predicts $(v_I^{a(2)},...,v_I^{a(L)})$ and $(v_I^{b(2)},...,v_I^{b(L)})$ conditioned on the observations of $\{QT_i\}_{I-S+1:I}$ and $\{TD_i\}_{I-S+1:I}$, with S being the time series sample size, i.e., the maximum number of timesteps that the model looks back in TAQ data history.

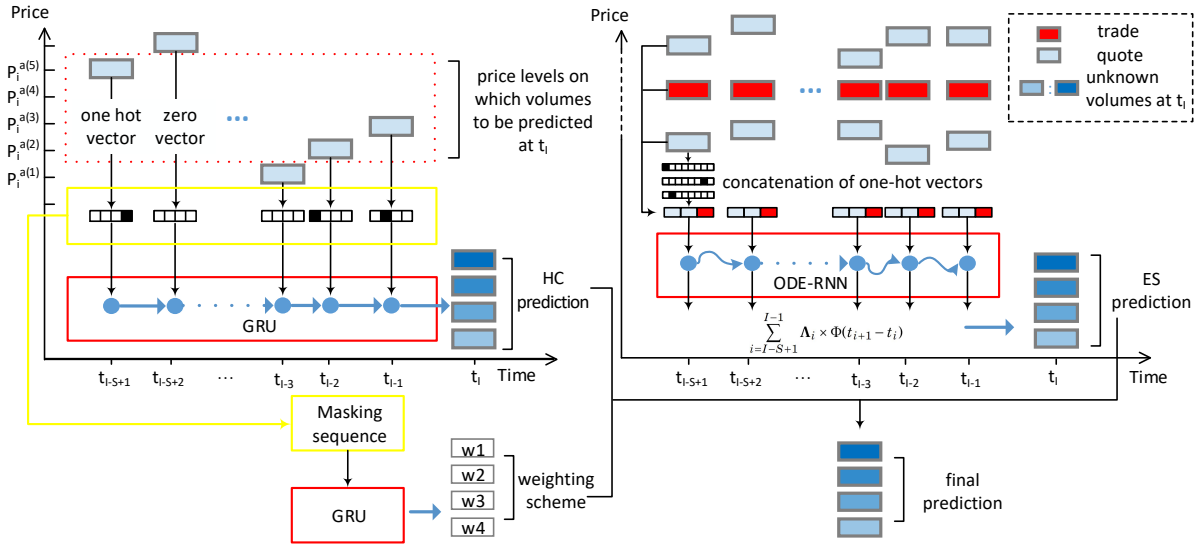


Figure 4.1: Structure of the LOBRM for the ask side, composed of a history compiler, an events simulator, and a weighting scheme module. The HC module equipped with a restricted reception field compiles prediction relevant history quote volumes to form a basis prediction; The ES module simulates the additional net order arrivals based on complete TAQ history to form an incremental prediction, using a continuous ODE-RNN; The WS module learns to adaptively combine two predictions based on the masking sequence of HC inputs, arriving at a final prediction.

4.3 The LOB Recreation Model

An overview of the model structure is presented in Fig. 4.1. As a common practice in exploiting the LOB, the ask side and bid side of the LOB are modelled separately. Here only the modelling of the ask side is illustrated, as the modelling of the bid side follows exactly the same logic. In this section, firstly the data standardisation and the encoding method used for TAQ data are presented, and then the three components of the LOBRM: the history compiler, the events simulator, and the adaptive weighting scheme are described in details.

4.3.1 LOB Data Standardisation

Data standardisation is necessary for the model's understanding of data of various numerical scales, especially for training and testing data that span across multiple trading days. As the updates in LOB time series are event-based and the time intervals between event arrivals are not regular, a standardisation method that considers this irregularity is needed. Thus, the time-weighted z-score standardisation is implemented on all LOB volumes, based on the fact that the LOB is a continuous dynamic system with uneven time intervals between updates. $\left\{v_i^{(l)}\right\}_{i=1}^{N}$ and $\{t_i\}_{i=1}^{N}$ are used to indicate volume trajectory on price level $l \in \{1,...,5\}$, and affiliated timestamps. N is the total number of LOB updates for training. Time-weighted mean and standard deviation

at a price level l are calculated as:

$$\mu^{(l)} = \frac{\sum_{i=0}^{N-1}(t_{i+1}-t_i)v_i^{(l)}}{t_N - t_0} \tag{4.1}$$

$$\sigma^{(l)} = \left(\frac{\sum_{i=0}^{N-1}(t_{i+1}-t_i)\left(v_i^{(l)}-\mu^{(l)}\right)^2}{t_N - t_0} \right)^{\frac{1}{2}} \tag{4.2}$$

From empirical observation, it is witnessed that the volume statistics on deep price levels $\left\{v_i^{(2-5)}\right\}_{i=1}^{N}$ have similar patterns, while those statistics deviate from volume statistics on the top price level $\left\{v_i^{(1)}\right\}_{i=1}^{N}$. In particular, the top level tends to have much lower mean and much higher standard deviation (e.g., see later in Table 4.1). Thus, the *top* and the *deep* levels are treated as two separate sets to standardise. As trade volumes $\left\{v_i^{tq}\right\}_{i=1}^{N}$ are discrete events and do not persist in time, the normal z-score standardisation method is used for trade data. To avoid lookahead bias, statistics are calculated without considering test data. Finally, using these statistics, LOBs for training, validation, and testing are standardised.

4.3.2 One-Hot Positional Encoding for TAQ Data

Sparse encoding in deep learning offers several advantages, particularly when dealing with high-dimensional data. It promotes computational and memory efficiency by focusing only on the active or relevant features, reducing the redundancy inherent in dense representations. This selective focus can lead to better model performance and generalisation, as the model learns to prioritise the most crucial features. Additionally, sparse representations, such as one-hot encoding for categorical data or bag-of-words for text, are intuitive and can simplify the handling of certain types of data in neural networks.

TAQ data contains multi-dimensional multi-modal information, including order type (*bid* or *ask*), price, and volume. While under the formulation of LOBRM, only order volumes at derived price levels (i.e., *deep* levels 2-5) are predicted. The one-hot positional encoding method is used, such that only volume information is encoded explicitly; while price is indicated by the position of non-zero element in the one-hot vector.

Take the encoding of an ask quote as an example. Conditioned on current best ask price $p_I^{a(1)}$ and best bid price $p_I^{b(1)}$ at t_I, the ask quote record $(p_{I-s}^{a(1)}, v_{I-s}^{a(1)})$ and bid quote record $(p_{I-s}^{b(1)}, v_{I-s}^{b(1)})$ at t_{I-s}, $s \in \{0,\ldots,S-1\}$ are denoted as:

$$\begin{cases} \mathbf{O}_{2k-1}^{aq}, \text{where } o_{k+sp_s^a} = v_{I-s}^{a(1)} \\ \mathbf{O}_{2k-1}^{bq}, \text{where } o_{k+sp_s^b} = v_{I-s}^{b(1)} \end{cases} \tag{4.3}$$

where $k \in R^+$, $sp_s^a = (p_{I-s}^{a(1)} - p_I^{a(1)})/\tau$ and $sp_s^b = (p_{I-s}^{b(1)} - p_I^{b(1)})/\tau$ indicating the distance between historical quotes to current quotes. $\mathbf{O}_{2k-1}$ is a one-hot vector with dimension $1 \times (2k-1)$; and o_{sp}

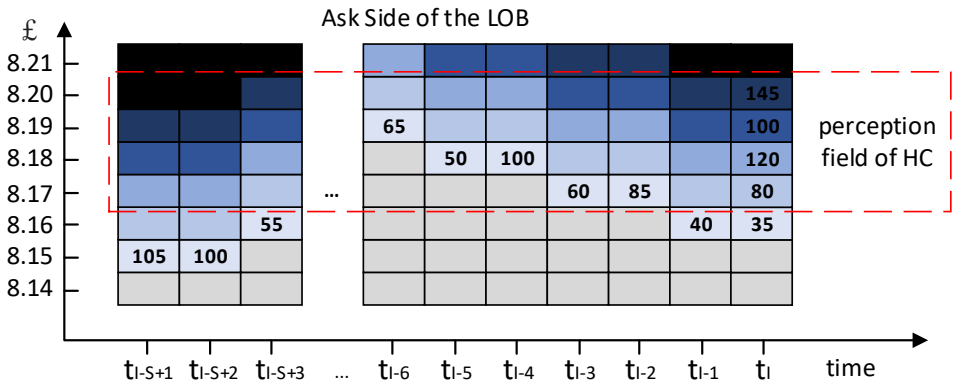


Figure 4.2: The perception field of the HC module. The HC module looks back in history to see how many orders were sitting at the target time's deep price levels (i.e., £8.17 to £8.20).

denotes the sp-th element of the vector. The value of k is chosen to cover more than 90% of past quote price fluctuations, relative to the current quote price. Here $k = 8$, which means historical quotes with relative price $[-7, +7]$ ticks are encoded into feature vectors. Then, a trade record $(p^{td}_{I-s}, v^{td}_{I-s}, d^{td}_{I-s})$, is represented as:

$$\begin{cases} \mathbf{O}^{td}_{2k-1}, \text{where } o_{p^{td}_{I-s} - p^{a(1)}_I} = v^{td}_{I-s} & \text{if } d^{td}_{I-s} < 0 \\ \mathbf{O}^{td}_{2k-1}, \text{where } o_{p^{td}_{I-s} - p^{b(1)}_I} = v^{td}_{I-s} & \text{if } d^{td}_{I-s} > 0 \end{cases} \tag{4.4}$$

Finally, those three features are concatenated into $(\mathbf{O}^{aq}, \mathbf{O}^{bq}, \mathbf{O}^{td})$ and are used as input into the model. It can also be a concatenation of four features, with one of the ask trade or bid trade being zero vector. Later, in experiment section, this sparse encoding method is shown to be achieving enhanced robustness in LOB volume prediction and mid-price movement prediction, comparing with the dense encoding method.

4.3.3 History Compiler Module

The HC module predicts LOB volumes from a historical perspective. It concentrates on historical quotes that are most relevant to current LOB volumes at deep price levels. More precisely, for an ask side model the volumes to be predicted at target time t_I are of prices $\{p^{a(1)}_I + \tau, \ldots, p^{a(1)}_I + (L-1)\tau\}$. Only historical ask quotes with price within this range are used as inputs into the HC module. As one-hot encoded ask quotes fall within the price range of $\{p^{a(1)}_I - k\tau, \ldots, p^{a(1)}_I + k\tau\}$, vectors need to be trimmed to remove verbose information and retain the most relevant information. Formally, a trimmed ask quote record $(p^{a(1)}_{I-s}, v^{a(1)}_{I-s})$ is represented as:

$$\begin{cases} \mathbf{O}_{L-1}, \text{where } o_{sp^a_s} = v^{a(1)}_{I-s} & \text{if } sp^a_s \in [1, L-1] \\ \mathbf{Z}_{L-1} & \text{otherwise} \end{cases} \tag{4.5}$$

where $\mathbf{O}_{L-1}$ is a one-hot vector of dimension $1 \times (L-1)$; $o_{sp^a_s}$ denotes the sp^a_s-th element of the vector; and $\mathbf{Z}_{L-1}$ denotes a zero vector of the same dimension. The intention of the HC module

is to look back in history to check how many orders were resting at price levels of interest at target time. As suggested by empirical observation that large portion of the LOB tends to persist in time [120], historical records of volumes on the deep price levels at the target time t_I might be indicative of the current volume information. Fig. 4.2 shows the perception field of the HC module. Thus, input feature vectors are manually trimmed to leave out verbose information. A discrete GRU unit is used to compile trimmed features and generate volume predictions which are used as the basis prediction derived from historical order dynamics on relevant price levels.

4.3.4 Events Simulator Module

The ES predicts the LOB from a dynamic perspective. As market participants, particularly algorithmic trading systems, submit and cancel orders at millisecond granularity, the arrivals of limit orders at different price levels are modeled as stochastic processes by the means of RNNs. Following prior stochastic modelling of the LOB, it is assumed that the occurrences of market events follow independent Poisson processes with time-varying arrival rates, and consequently their aggregation effect on net order arrivals still follows a pooled inhomogeneous Poisson process. If an RNN structure is used to iteratively receive encoded TAQ features at every timestep, its latent state can be deemed as reflective of market microstructure condition over a short historical time window. A multi-layer perceptron (MLP) can then be used to decode latent states directly into vectors representing net order arrival rates at each price level. Additionally as TAQ sequences are irregularly sampled time series in the sense that time intervals between timestamps are uneven, here a continuous ODE-RNN module instead of a discrete RNN is used to receive iterative inputs. This inclusion of temporal information within the model structure itself is shown to be robust and can improve prediction accuracy.

Following this terminology, at discrete input time t_{I-s}, the hidden state of ODE-RNN is updated from $\mathbf{h}_{I-s}$ to $\mathbf{h}'_{I-s}$with a GRU unit:

$$\mathbf{h}'_{I-s} = \text{GRU}(\mathbf{h}_{I-s}, \mathbf{x}_{I-s}) \tag{4.6}$$

Then a separate neural network f_θ parameterised by θ is used to describe the evolution of the hidden state during time interval $[t_{I-s}, t_{I-s+1})$, and the hidden state before next input can be obtained by using an ODE solver:

$$\frac{d\mathbf{h}(t)}{dt} = f_\theta(\mathbf{h}(t), t) \text{ where } \mathbf{h}(t_{I-S+1}) = \mathbf{h}_0 \tag{4.7}$$

$$\mathbf{h}_{I-s+1} = \text{ODEsolver}(f_\theta, \mathbf{h}'_{I-s}, \Phi(t_{I-s}, t_{I-s+1})) \tag{4.8}$$

where Φ is a smoothing function, usually a logarithm function, to make the differences in time intervals less abrupt, in order to achieve a more robust result.

The vector of net order arrival rates $\mathbf{\Lambda}_{I-s} = [\lambda_{I-s}^{a(2)}, ..., \lambda_{I-s}^{a(L)}]$ for different price levels at time t_{I-s} is directly derived from the latent state h_{I-s} using MLP layers as:

$$\mathbf{\Lambda}_{I-s} = \text{MLP}(\mathbf{h}_{I-s}) \tag{4.9}$$

After acquiring the trajectory of $\mathbf{\Lambda}$ at all trade times over the defined length of timesteps, the accumulated order volumes between $[t_{I-S+1}, t_I]$ can be calculated as:

$$\sum_{i=I-S+1}^{I-1} \mathbf{\Lambda}_i \times \Phi(t_{i+1} - t_i) \tag{4.10}$$

The terminology of the ES prediction is straightforward: on the basis of the HC prediction from a historical perspective, it provides an incremental prediction by accounting for the extra net order arrivals during the prediction period, by treating the arrivals as stochastic processes.

4.3.5 Weighting Scheme Module

The WS module is designed to combine the predictions from the ES and HC modules into a final prediction. It follows the intuition that if the HC prediction for a particular price level is reliable from a historical perspective, a higher weight will be allocated to it, and *vice versa*. If quote history for a target time price level is both abundant and recent, the information provided on current LOB volumes is perceived as more reliable. Take Fig. 4.2 as an example, the volume information provided on $p^{a(2)}$ and $p^{a(3)}$ by past quote history is more abundant than other price levels; and the information provided on $p^{a(2)}$ is more recent than $p^{a(3)}$. The abundancy and timing of historical quotes are denoted by the masking sequence of HC inputs. Formally, the mask of an ask side HC input is represented as:

$$\begin{cases} \mathbf{O}_{L-1}, \text{where } o_{sp_s^a} = 1 & \text{if } sp_s^a \in [1, L-1] \\ \mathbf{Z}_{L-1} & \text{otherwise} \end{cases} \tag{4.11}$$

A GRU unit is used to receive the whole masking sequence to generate a weighting vector of size $1 \times (L-1)$ to combine predictions from ES and HC modules.

To summarize, the model structure of the WS module is identical to that of the HC module, employing a discrete GRU. The only distinction between them is that the input used in the WS module is the 0/1 masking of the input in the HC module. This masking sequence indicates the timing and relative price of a historical quote within a window size of S. The output of the WS module is of size $1 \times (L-1)$, with each entry limited to the value range of $[0, 1]$, indicating the extent to which the model should consider the prediction from the HC module compared with the one from the WS module at each price level. Importantly, the allocation of weights is adaptive (not rule-based) and learned from the data, with the mechanism designed based on empirical observation that volume at a specific price level tends to persist at short time scales (i.e., several seconds).

Table 4.1: Volume statistics before standardisation, showing time-weighted mean volume and standard deviation on the top level and deep levels (levels 2 to 5).

	MSFT		INTC		JPM	
	Bid	Ask	Bid	Ask	Bid	Ask
Top	119.7/89.1		127.1/98.3		32.7/38.5	
Deep	189.6/51.7	192.7/53.4	185.6/62.3	178.9/51.4	61.3/23.9	63.4/40.5

4.4 Experiments

4.4.1 Data Preprocessing

Experiments are conducted on the LOBSTER dataset, providing real world LOB event stream data of five days' length for three stocks, ticker symbol INTC (*Intel*), MSFT (*Microsoft*), and JPM (*J.P. Morgan*). The dataset contains only top five price levels information on both sides of the book. On average there are 0.45 million event updates per trading day per stock. Following Blanchet et al. [120], only unique time points on which trades happen are selected for LOB recreation. As a common practice to reduce noise, the LOB data during the first half-hour after market open and the last half-hour before market close is removed. TAQ data is directly extracted from the LOB and its affiliated message book. To alleviate the effect of outliers, all volume numbers are divided by 100 and winsorised by the range $[0.005, 0.995]$. The processed data distribution is shown in Fig. 4.3, reflecting the distributions for order price and volume. It can be seen that order volumes generally follow Gaussian distributions, in accordance with the z-score standardisation method. Then, the standardisation method proposed in Section 4.3.1 is applied to the cleansed LOB data. Data statistics are illustrated in Table 4.1. Parameters are set as $S = 100$ and $k = 8$. After testing $S = \{50, 100, 150\}$, $S = 100$ was found to achieve the best performance considering both accuracy and time efficiency; while k is selected to cover more than 90% of observed price movements in window of size S. For training setting, the first three days' data is used for training, the fourth day's data for validation, and the fifth day's data for testing.

4.4.2 Model Comparison

In this section, training results on predicting the LOB volumes on deep price levels are obtained by using mainstream regression and machine learning methods: (1) Support vector machine regression with linear kernel (LSVR); (2) Ridge regression (RR); (3) Single layer feedforward network (SLFN); (4) XGBoost regression (XGBR); (5) Attention neural network (ATTN). For models that cannot handle high dimensional time series features, a PCA dimensionality reduction is applied on the flattened features. The performance of LOBRM is compared with either discrete RNN or continuous RNN for the ES module: (1) GRU; (2) RNN-Decay, a continuous RNN variant with exponential decaying mechanism in latent state; and (3) ODE-RNN, the proposed module. Two criteria are used: (1) L1 loss on test set. As all labels are standardised into z-score, the loss

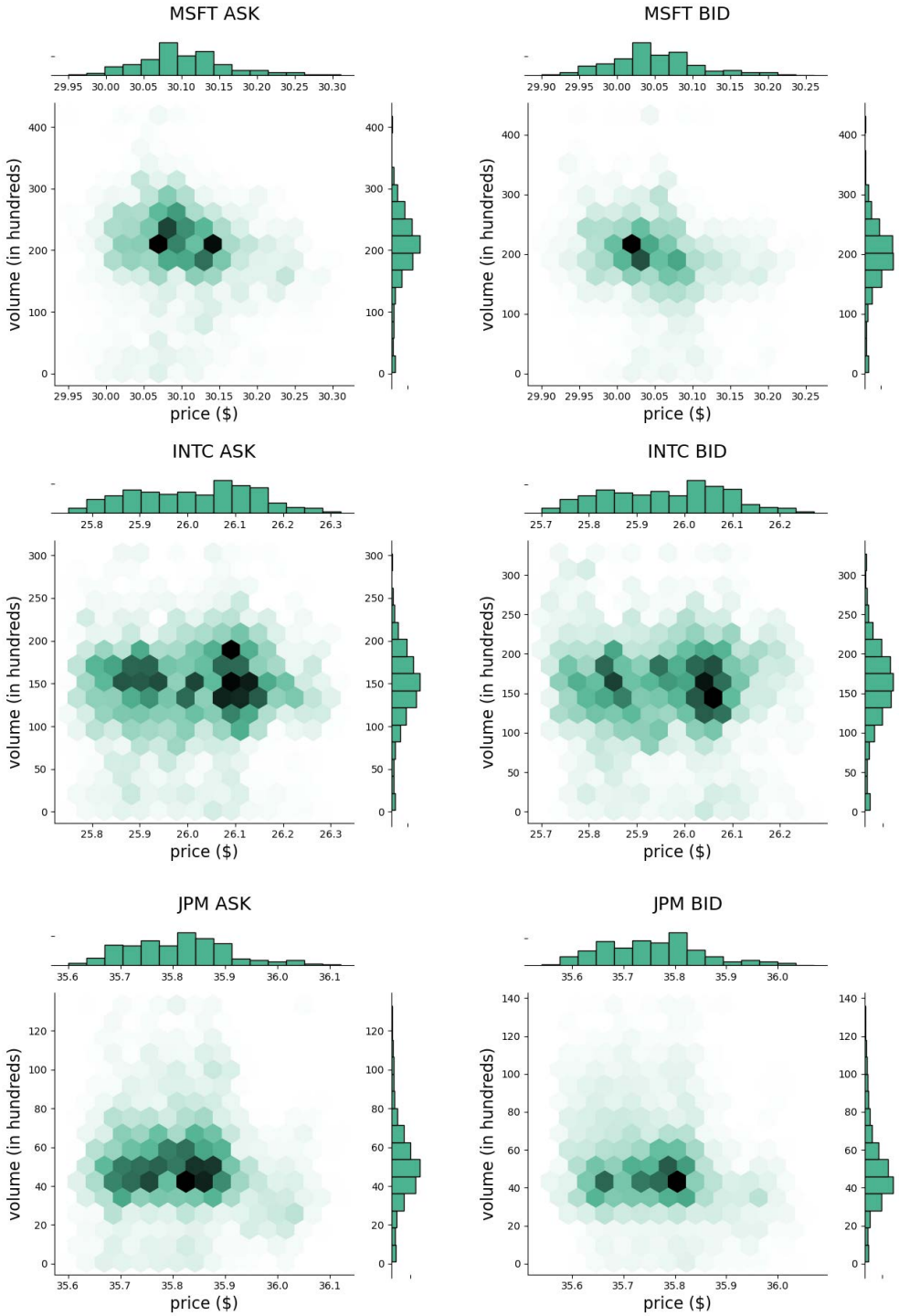


Figure 4.3: 2-D distribution of the quotes in the LOBSTER dataset after winsorisation. Only one day data (on 06/26/2026) is selected for visualization as an example. It can be seen that the quote volume distribution generally follows a Gaussian distribution, confirming with the z-score standardization method for quote volume data. In comparison to quote volume, quote price distributions tend to deviate from a Gaussian form.

Table 4.2: Model comparison. Criteria shown in format: test loss/R-squared; all numbers are in 1e-1. The lowest test loss and highest R-squared in each set are underlined.

Model	MSFT		INTC		JPM	
	Bid	Ask	Bid	Ask	Bid	Ask
LSVR	6.75/0.94	6.77/1.13	5.20/1.23	5.51/0.78	6.12/0.96	3.30/1.29
RR	6.51/0.87	6.52/1.18	5.11/1.17	5.67/0.76	6.32/1.00	3.33/1.15
SLFN	6.67/0.93	6.72/1.11	5.30/1.08	5.79/0.74	6.12/1.01	3.31/1.26
XGBR	6.80/0.86	6.81/1.06	5.38/0.90	6.05/0.80	6.51/1.06	3.53/0.92
ATTN	6.59/0.86	6.68/0.91	5.25/0.96	5.49/0.79	6.28/0.94	3.32/1.12
LOBRM [GRU]	6.42/1.33	6.49/1.58	5.28/1.30	5.51/0.96	6.03/1.39	3.22/1.50
LOBRM [Decay]	6.45/1.34	6.51/1.52	5.15/1.32	5.57/0.93	6.00/1.35	3.20/1.61
LOBRM [ODE-RNN]	6.35/1.31	6.38/1.72	5.26/1.28	5.46/0.99	6.11/1.67	3.20/1.85

indicates the multiple of standard deviations between prediction and ground truth; (2) R-squared, calculated on the test data the same way as in Blanchet et al. [120], to enable us to perform a comparison with the existing literature.

Model Specifications The ES module consists of a GRU with 64 units for discrete updating of the latent state, a three layered MLP with 64 units and Tanh activation for deriving the derivative of the latent state, and a three layered MLP decoder with 64 units and Tanh activation. The HC module consists of a GRU with 64 units, and a three layered MLP decoder with 64 units and Tanh activation. The latent state size is set at 64 in ES and HC modules. The WS module consists of a GRU with 16 units, and an one-layered MLP decoder with 16 units and Sigmoid activation. The latent state size is set at 16 in the WS module. Batch size is set as 512; L1 loss is used as the loss function; and all models are trained for 50 iterations, with a learning rate of 1e-3; RMSprop is used as the optimiser. Model parameters are chosen based on the lowest loss on the validation set. All experiments are repeated five times to get the mean of criteria.

Table 4.2 presents evaluation results. Judging from both criteria, it is observed that the LOBRM family outperforms traditional linear and non-linear models by effective ensembling of RNNs. This indicates that the recurrent structure of RNNs can facilitate the model's capability in explaining temporal variance in LOB volume. Further, LOBRMs with a continuous RNN module exhibit superior performance over those with a discrete RNN module considering both criteria (11 out of 12). LOBRM [ODE-RNN] achieves the lowest average L1 test loss and highest average R-squared (8 out of 12), indicating that incorporating temporal information in latent state dynamics can help the model to capture its dependence on time. Fig. 4.4 shows volume heatmaps for the real LOB, the simulated LOB (by stacking consecutive predictions), and the absolute error between those two, during an one-hour slot for INTC test data. As observed, the predictions in general mimic the real LOB, and the majority of errors are below 50×10^2 units. Thus, in the following experiments, the LOBRM [ODE-RNN] is used as the main model.

As shown in Table 4.2, test losses across six sets of experiments fluctuate in the range

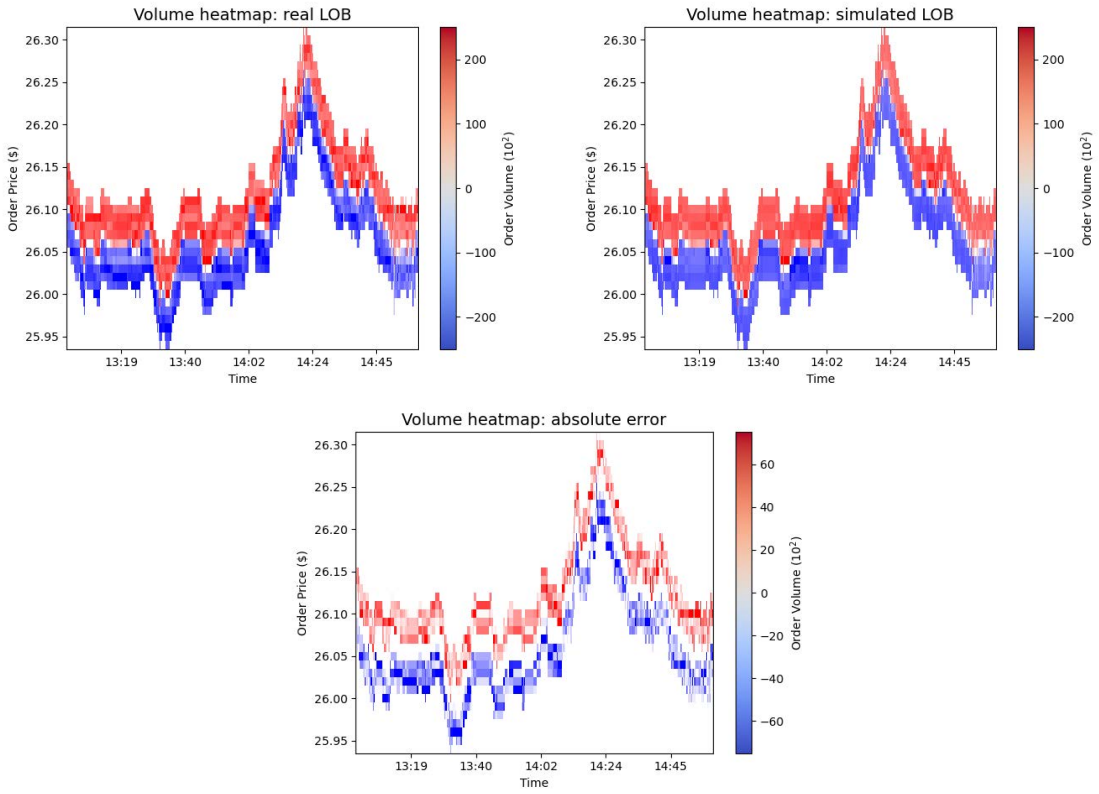


Figure 4.4: Volume heatmap for the real LOB, the simulated LOB, and the absolute error. Volumes on the ask side are in positive numbers and the ones on the bid side are in negative numbers.

[0.320, 0.638] for LOBRM [ODE-RNN]. To test whether this fluctuation results from order volume volatility, the correlation of (hourly) test L1 loss across all experiments against the (hourly) standard deviation of standardised volume resting at deep price levels is calculated (see Fig. 4.5). It is found that the correlation between hourly L1 loss and volume volatility is significant (p-value<0.01) with a R-squared value higher than 0.9. Thus, it can be concluded that LOB prediction accuracy is negatively related to volume volatility. This conclusion is in accordance with intuition, as a more volatile market is inherently harder to predict.

In Blanchet et al. [120], the R-squared for daily average prediction against ground truth lies in the range of [0.81, 0.88]. However, in comparison, our R-squared values presented in Table 4.2 are regressed against the ground truth with no averaging procedure. As the model only has one-day length of data for testing, it is unable to generate daily average volume prediction for R-squared calculation. When volume predictions are averaged into hourly frequency, the value of R-squared across 6 experiments ranges in [0.41, 0.87] with an average value of 0.66. It is found that our results are comparable to the existing literature, even though those results are not measured under identical conditions.

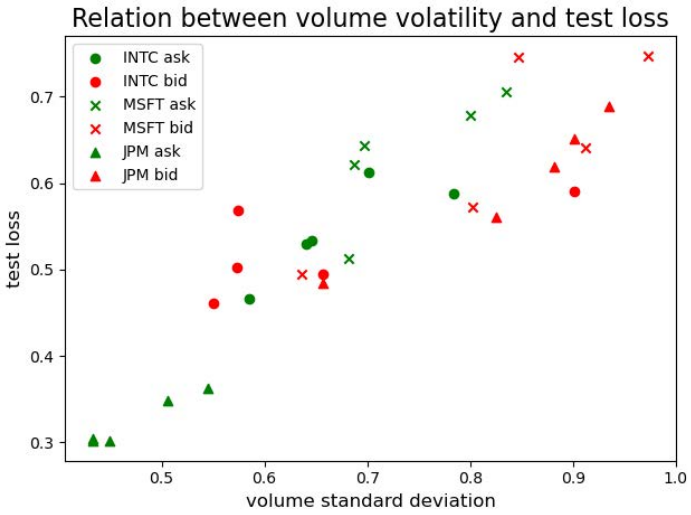


Figure 4.5: Hourly test loss plotted against hourly volume standard deviation.

Table 4.3: Ablation study: test loss. Lowest loss is underlined.

	MSFT		INTC		JPM		Avg.
	Bid	Ask	Bid	Ask	Bid	Ask	
HC	6.91	7.32	5.38	5.54	6.37	3.35	5.81
ES	6.52	6.44	5.21	5.60	6.19	3.31	5.55
HC+ES	6.39	6.48	5.38	5.51	6.00	3.21	5.50
HC+ES+WS	6.35	6.38	5.26	5.47	6.11	3.20	5.46

4.4.3 Ablation Study

An ablation study is conducted to demonstrate the effectiveness of module ensembling. Four experiments are conducted: HC, using only the history compiler; ES, using only the events simulator; HC+ES, which includes the history compiler and the events simulator, using a predefined weight to combine outputs; and HC+ES+WS, which is the full LOBRM with adaptive weighting scheme.

Results are shown in Table 4.3. It is found that predictions from the HC alone have the highest test error, as the inputs it receives are trimmed to contain only the most relevant data for current LOB volume prediction, and no temporal dynamics is considered. ES module, by receiving complete TAQ data input and modelling market events as a stochastic process using a continuous RNN, achieves a lower error than HC module and is the dominant module that contributes to prediction accuracy. Combining both HC and ES modules, either using a predefined or adaptive weight, achieves the best performance, which suggests that the predictions from HC and ES are complementary and can be effectively combined to gain a higher accuracy prediction. By comparing models with and without the WS module, it can be seen that using a learnable adaptive weighting method can further improve model performance.

Table 4.4: Explicit versus sparse encoding: predicting LOB from TAQ

	MSFT		INTC		JPM		Avg.
	Bid	Ask	Bid	Ask	Bid	Ask	
Explicit	6.99	8.03	5.49	6.10	6.74	3.52	6.15
Sparse	6.35	6.38	5.26	5.47	6.11	3.20	5.46

4.4.4 Superiority of Sparse Encoding for TAQ

LOB Volume Prediction In sparse encoding, only volume information in TAQ is encoded explicitly and price information are embedded implicitly by positions of non-zero elements in one-hot vectors. In explicit encoding, information including price, volume, and trade direction are encoded directly as non-zero elements in feature vectors. Here the ES module in LOBRM [ODE-RNN] is used as the main model and the model prediction accuracy is tested when these two different encoding methods are used. The full model is not chosen as explicitly encoded input cannot be trimmed so HC module is not used. Results are shown in Table 4.4. It is found that the model with sparse encoding achieves lower test loss error in all experiments. The average test loss for the model with sparse encoding is 11% lower than the model with explicit encoding.

Mid-Price Movement Prediction In this part, the sparse encoding method is compared with the convolution method proposed in Zhang et al. [14] for quotes data in the task of stock mid-price movement prediction. On the basis of explicit encoding, the convolution encoding method applies two convolution layers with filters of size $[1 \times 2]$ and stride $[1 \times 2]$ on quote data. This structure first convolutionalises price and volume information at ask and bid sides respectively and then convolutionalises two sides' information together. A similar task presented in Zhang et al. [14] is approached but the model structure is simplified to lay concentration on the prediction accuracy variation brought by different encoding methods. The length of time series samples is set as $S = 50$. For sparse encoding, $k = 5$. For convolution encoding, The setting of 16 $[1 \times 2]$ kernels with stride $[1 \times 2]$ is used followed by LeakyReLU activation. Features are standardised using min-max or z-score standardisation. Encoded data are passed to an MLP with ReLU activation. A GRU unit is used to receive iterative inputs and the final latent state is connected with an MLP with Softmax activation to generate a possibility distribution over three labels $\{down, same, up\}$. The model is tested on MSFT five-day dataset, with first three days for training, the fourth day for validation, and the fifth day for testing. The model runs a rolling average of five timesteps on labels to alleviate label imbalance [124], with 29%, 40%, and 31% for *up*, *same*, and *down*. The model is trained with cross entropy loss for 100 iterations. Model parameters are chosen based on the lowest loss on the validation set.

Results are shown in Table 4.5. It can be seen that the sparse encoding method with two different standardisation methods has superior performance in the task of price trend prediction, compared with convolution encoding either with or without price information. Thus, the conclu-

Table 4.5: Explicit versus sparse encoding: predicting future mid-price movements

	Sparse (implicit price)	Convolution (explicit price)	Convolution (no price)
Min-max	54.2%	52.8%	53.4%
Z-score	56.5%	54.6%	54.5%

Table 4.6: LOBRM [ODE-RNN] test loss against training size.

	MSFT		INTC		JPM		Avg.
	Bid	Ask	Bid	Ask	Bid	Ask	
Day3	6.67	6.68	5.29	5.65	6.19	3.32	5.63
Day2+3	6.45	6.54	5.20	5.40	6.17	3.33	5.52
Day1+2+3	6.28	6.15	5.33	5.64	6.18	3.32	5.48

sion is drawn that the sparse encoding method for TAQ data can not only benefit the task of LOB volume prediction, but also other tasks including stock price trend prediction.

4.4.5 Is the Model Well-Trained?

As financial time series suffer from stochastic drift, in the sense that the distribution of data is unstable and tends to vary temporarily, large amounts of data is needed for model training. For example, the LOB used in Blanchet et al. [120] and Sirignano and Cont [103] is of one month's and seventeen month's length. Here, for all six sets of experiments (3 stocks × 2 sides), three days' LOB data is used for training, one day's data for validation, and one day's data for testing. It is tested that whether this amount of training data is abundant enough for out-of-sample testing.

As the dataset contains five consecutive trading days' LOB data, the fourth day's and fifth day's LOB data are left out for validation and testing. The model first uses the third day's data for training and then the second and the first day's historical data are iteratively added in to observe how the validation and testing loss change. Results are shown in Table 4.6. It can be observed that there is a downward tendency in average test loss as more historical data is used for training. This phenomenon suggests that the influence of stochastic drift may be alleviated by exposing the model to more historical samples. Therefore, even lower out-of-sample errors would be expected if more historical data can be used for training.

4.5 Conclusions

This chapter presented the LOB recreation model (LOBRM) that predicts deep level LOB volume information from TAQ data. It is the first deep learning model that bridged the gap between the TAQ and LOB data, achieving a simulation of the LOB for *large-tick* stocks from a static view. The LOBRM contains three components: a history compiler, an events simulator, and an adaptive weighting scheme. We have demonstrated accuracy is improved by encoding irregularly sampled

TAQ data into a continuous latent state using an ODE-RNN, compared with using discrete RNN variants. Through an ablation study, we found that even though the events simulator plays a dominant role in LOB prediction accuracy, the combination of all three components further improves accuracy. Further empirical findings were that: (1) LOB volume prediction accuracy is negatively related to volume volatility; (2) sparse one-hot positional encoding of TAQ data can benefit manifold tasks; and (3) evident shows that the influence of stochastic drift can be alleviated by increasing the number of historical samples.

The LOBRM enables us to create a synthetic LOB at no extra cost from only TAQ data. In cases where either the historical LOB records are incomplete, or online streaming of the LOB data is prohibitively expensive, the LOBRM can be a valuable tool for practitioners and researchers alike. On the other hand, its shortcomings are also obvious. In this thesis the LOBRM is called a static model in the sense that it does not consider the dynamics of the LOB itself. For instance, volume information on two adjacent LOB snapshots should be very alike, as the difference between them results from only a few order submissions or cancellations at each price level. Instead, in the LOBRM this dynamic is not considered, and the difference between predictions for two adjacent LOB snapshots only results from difference in input TAQ sequence. Also, the recreated LOB is of a lower updating frequency compared with a real LOB, owing to the fact that the TAQ it aligns with is in nature downsampled. Thus, a more realistic event-based modelling of the LOB, in which adjacent LOB variances are directly linked to order streams, is needed.

CHAPTER 5

MODELLING THE LOB FROM EVENT STREAM: A DYNAMIC VIEW

The content to be presented is a generalisation of previously published work [125].

5.1 Introduction

Financial markets, and in particular, LOB markets, are intricate systems that operate within a web of interconnected events and decisions. These systems are driven by a multitude of events, such as order submissions, modifications, and cancellations, each contributing to the overall market dynamics. For traders, institutional investors, and market regulators, the capability to accurately predict and comprehend these events is not just a matter of strategic advantage but also of ensuring market stability and efficiency. As indicated by empirical analysis on the LOB, it has been established that the sequence and timing of events in a LOB market can be effectively encapsulated using stochastic point processes. These mathematical constructs, equipped with intensity functions, serve as a robust framework, enabling researchers and analysts to quantify and predict the frequency of event arrivals over a specified duration.

Basic models, exemplified by the Poisson process [8], operate under the assumption that intensity rates remain constant and are uninfluenced by past events. However, as any seasoned market participant would attest, real world market dynamics are replete with feedback loops and inter-dependencies. Recognising this complexity, advanced models, such as the Hawkes process [112], have been applied. These models are designed to account for the intricate interplay of events, where the intensity rate of a particular event is dependent not only on its own historical occurrences but also on the history of other correlated events. By capturing this interdependence, these models offer a more granular and holistic perspective of market events, transcending the limitations of static intensity rates. Given their ability to mirror real world complexities,

stochastic point processes have emerged as an indispensable tool, finding widespread adoption in the modelling and analysis of LOB market behaviors.

The last decade has witnessed a paradigm shift in modelling techniques, driven by the rise of deep learning. This computational revolution has paved the way for the development of the neural point process, a concept that synergises the foundational theories of stochastic point processes with the superior computational capability of neural networks [16, 126]. By combining the robustness and interpretability of traditional models with the flexibility and precision of neural networks, this hybrid approach offers a powerful tool for modelling multi-variate point processes. Its efficacy is already evident across diverse domains, from healthcare [127] to e-commerce [128] and digital communication [129]. Within the context of LOB markets, the neural point process provides a refined modelling approach. By categorising events types in a LOB market according to the order type (i.e., submission or cancellation) and the side it is affiliated with (i.e., ask or bid), neural point process family models can be easily adapted to learn the patterns embedded in order streams. Another unique characteristic of modelling the LOB event stream in this manner is that it provides us with a validated way to stochastically sample (e.g., using the thinning algorithm [130]) the most likely next event type and time according to the evolution of the intensity function. Further, by combining the iteratively generated events with affiliated price and volume information sampled from empirical distributions, an alternative method of LOB simulation can be achieved.

This chapter presents the state dependent parallel neural Hawkes process (sd-PNHP), a systematic attempt to utilise neural point process family models in LOB market event stream prediction. The research further extends the topic from event type prediction to realistic LOB data simulation, to enrich current methodologies in generating simulated LOB data. The main contributions are:

- The parallel continuous-time long short-term memory (PCT-LSTM) module is proposed to model the intensity rates of different types of events in LOB markets. In its core, K (the number of event types) continuous-time long short-term memory (CT-LSTM) units are stacked in parallel to encode the whole event stream sequence into separate exponentially decaying latent states. Unlike the original neural Hawkes process, which uses a single CT-LSTM unit for all events, the parameters that control the evolution of latent states are tailored to each event type and are no longer shared.

- An event-state interaction mechanism is introduced, in which the current market state indicator influences the evolution of intensity rates and, in turn, the type of each event arrival influences the state transition probability. This mechanism is shown to enhance prediction accuracy for event type and time, while also allowing efficient sampling of the event state pairs for LOB stream simulation.

- A minimal LOB simulation is attempted on the basis of the well-trained model. By implementing the thinning algorithm, the minimum time until the next event arrival and the most likely event type at that time are stochastically generated from a model that has been trained on historical event stream patterns. Simulating an event stream of arbitrary length is then made possible by iteratively incorporating new predictions as input to the next iteration.

In the experiment section, the performance of the proposed sd-PNHP is compared with other mainstream stochastic point process models and deep learning models. Discussions are made on the influence of the event-state interactive mechanism, and the choice of loss functions on model's performance. Finally, a minimal LOB simulation is attempted. The Sobol's global sensitivity analysis is implemented to investigate how some of the stochastic properties of the simulated LOB is sensitive to various input factors. Experiments demonstrate the model's potential as a competitive alternative for LOB prediction and realistic LOB simulation.

5.2 Problem Description

Given LOB market event stream data extracted from the message book data, samples are arranged as a set of event sequences $\{C^1,\dots,C^J\}$. $C^j=(e_1^j,\dots,e_I^j)$, $e_i^j=(t_i^j,x_i^j)$ in which $t_i^j\in\mathbb{R}^+$ and $x_i^j\in\{0,1,2,3\}$ respectively represent the time and event type of the i-th arrival of the j-th event sequence. Event arrivals are categorised into four classes: submission of order at the bid side ($x=0$), cancellation of order at the bid side ($x=1$), submission of order at the ask side ($x=2$), and cancellation of order at the ask side ($x=3$). The state sequence coupled with C^j is denoted as $S^j=(s_1^j,...,s_I^j)$. The event stream for a specific financial asset during each day is manually sliced into sub-streams of equal length I, using the rolling window approach of step size one. Given sequences C^j and S^j of length I, the model receives the first $I-1$ events as input and makes a prediction on the I-th event type x_I^j.

5.3 The State Dependent Parallel Neural Hawkes Process

This section presents the sd-PNHP for LOB event type and arrival time prediction. The proposed sd-PNHP differs from the traditional neural Hawkes process in its unique event-state interactive mechanism, and a stacked PCT-LSTM design. A discussion on the choice of the loss function that maximise the logarithm likelihood of the samples is also illustrated. Together, sd-PNHP provides a novel deep learning based stochastic modelling methodology for LOB event stream.

5.3.1 The Event-State Interactive Mechanism

One distinct characteristic of event streams in a LOB market is that the arrival of events not only depends on the event history, but is also influenced by the current market state [19, 20]. For

example, at any given time, the next event type is influenced by whether the current bid-ask spread is large or small, and whether or not the aggregate volume of orders is greater on the ask side or bid side. These incoming events can then reshape the LOB's structure by either adding or removing orders. This dynamic creates a feedback loop that can impact the market's current state. Recognising and incorporating this interplay between events and states in modelling can potentially enhance prediction accuracy.

In a neural Hawkes process, by referring Eq.3.11 to Eq.3.24, the influence of past event history laid on current event intensity rate during time interval $(t_i, t_{i+1}]$ is no longer additive. First, a neural network is used to iteratively encode history information into a hidden state, and the intensity rate is decoded from such a latent state when making predictions. Second, in order to incorporate time information, the exponential decaying mechanism is laid upon the latent state instead of the intensity rate directly. Considering the aforementioned event-state interaction mechanism, the parameters of the CT-LSTM at i-th discrete inputs update in a same way as in a neural Hawkes process apart from a modification to Eq.3.12:

$$\tilde{\mathbf{h}}_i^- = \text{concat}((\tilde{\mathbf{x}}_i + \tilde{\mathbf{s}}_i), \mathbf{h}_i^-) \tag{5.1}$$

where $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{s}}_i$ are embedded event and state input at time t_i. Now, the updates of the hidden state at discrete input timesteps are dependent on both the arrived event type and the market state at the arrival time.

By considering the feedback effect of arrived event on current market state in a stochastic point process [131], after event e_i happens, the state transition follows the distribution:

$$P(s_i \mid x_i, F_{i-}^{X,S}) \sim \phi_{x_i, s_{i-1}} \tag{5.2}$$

in which $F_{i-}^{X,S}$ indicates event stream and state history just before time t_i. $\phi_{x_i, s_{i-1}}$is a discrete distribution when the event type at time t_i is x_i and the state just before time t_i is s_{i-1}. Under such a formation, the intensity rates of events are influenced by the state, and the state transition thereafter is in turn influenced by which type of event arrives at the moment. A market condition indicator similar to Cartea et al. [9] is used, which is calculated as $I = (v^{b(1)} - v^{a(1)})/(v^{b(1)} + v^{a(1)})$ and the state is denoted as:

$$s = \begin{cases} 0 & I \in [-1, -\theta) \\ 1 & I \in [-\theta, \theta] \\ 2 & I \in (\theta, 1] \end{cases} \tag{5.3}$$

Such an *imbalance* indicator describes whether the buying or the selling side for an asset takes dominance. If the order volume on the ask side is much more than the bid side, the market is an *ask market* and the price tends to go downwards; Likewise if the volume on the bid side is

much more than the ask side, the market is a *bid market* and the price tends to go upwards; If the volume on both sides are roughly balanced, it is called a *balanced market* and fluctuation in price tends to have neutral direction.

5.3.2 Stacking of CT-LSTM Units

In the original neural Hawkes process, a single CT-LSTM unit is used to encode input and time information into the latent state with decaying kernels. The intensity rates for all types of events are decoded from the single latent state. Parameters related to encoding time information (i.e., c, $\bar{c}$, δ) are shared across all event types, and the expression of the decaying mechanism for different event type intensity rates is bounded.

With the aim of enriching the expression of latent state evolution for different types of events, K units of CT-LSTM are stacked in parallel to form a PCT-LSTM module for encoding input and time information. Each unit governs the evolution of the intensity rate for different event types. At the same time, to enable parameter sharing across different units, the mechanism used to update hidden states at discrete input timesteps is modified (shown in Eq. 3.12). The decayed hidden states of different units together with inputs are first passed through a common linear fully connected layer, and the outputs are then used as hidden states in respective units:

$$\tilde{\mathbf{h}}^{-}_{i,1},...,\tilde{\mathbf{h}}^{-}_{i,K} = \sigma_{tanh}(\mathrm{LC}_h(\mathrm{concat}((\tilde{\mathbf{x}}_i + \tilde{\mathbf{s}}_i), \mathbf{h}^{-}_{i,1},...,\mathbf{h}^{-}_{i,K})) \tag{5.4}$$

It is found in experiments that this shared linear layer at discrete input timesteps across stacked units can effectively improve prediction accuracy, with an improvement of approximately 2% in terms of type prediction accuracy compared with the model in which this parameter sharing layer is absent. After the hidden state is updated, the intensity rate for event type k in a single CT-LSTM unit module and a stacked PCT-LSTM module are respectively indicated as:

$$\lambda_k(t) = \mathrm{D}(\mathbf{h}(t))[k] = \mathrm{D}(\mathrm{CTLSTM}(S, X, t))[k] \tag{5.5}$$

$$\lambda_k(t) = \mathrm{D}_k(\mathbf{h}_k(t)) = \mathrm{D}_k(\mathrm{CTLSTM}_k(S, X, t)) \tag{5.6}$$

where D is the decoder for deriving intensity rates, consisting of a linear layer and Softplus activation. For comparison, Eq.5.5 uses the k-th element of the intensity rate vector derived from a single CT-LSTM unit for type-k event (as used in the original neural Hawkes process); Eq.5.6 uses the output of the k-th unit of the PCT-LSTM module to derive the intensity rate for type-k event specifically. The general framework of the proposed sd-PNHP model is illustrated in Fig. 5.1.

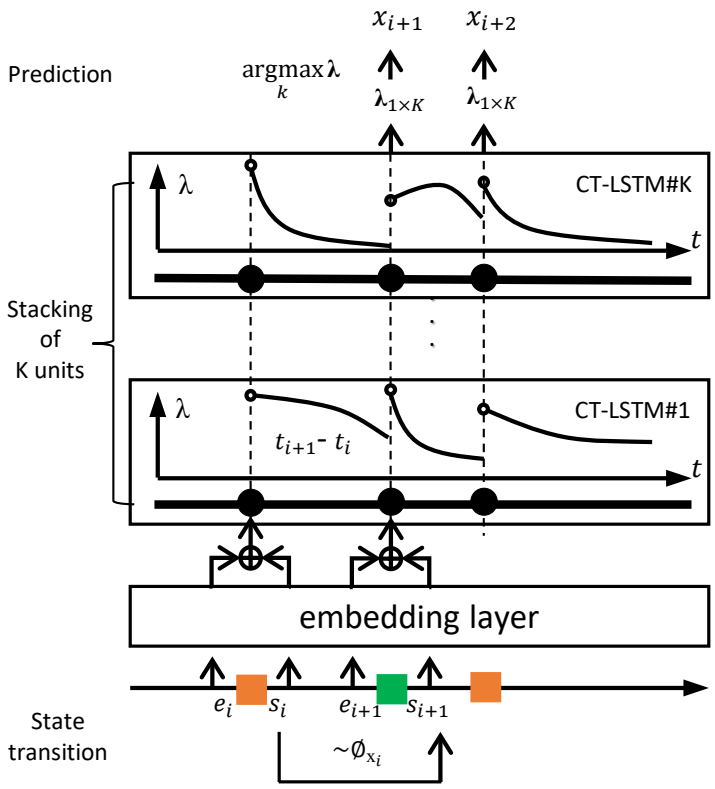


Figure 5.1: Structure of the sd-PNHP model. Event stream and state stream are first passed through an embedding layer, and the summed results are used as input into the PCT-LSTM module, a stacking of K CT-LSTM units each governs the evolution of the intensity rate curve for a certain type of event. At each known timestep where event arrives, the event type predicted is the event with maximum intensity rate; At the same time, state transition is dependent on the last arrived event, according to a distribution ϕ_{x_i}.

5.3.3 Loss Function

The primary goal of training a neural Hawkes process is to maximise the likelihood of the observed event sequence given model parameters. For the ease of gradient computation, the sum of logarithm probability of $p(t)$ is used instead in deriving the loss function. The logarithm probability for an event sequence C can be computed as:

$$L_1 = \sum_{i:t_i \leq T} \log \lambda_{x_i}(t_i) - \int_0^T \lambda(s) ds \tag{5.7}$$

of which the proof can be found in Section 3.3. It is termed L_1 the Hawkes loss term. By maximising Eq. 5.7, the probability that event x_i^j happens at time t_i^j in sequence C^j is maximised. Eq. 5.7 maximises the intensity function at event arrival times and minimises the function at non-event time. One thing to notice is that even though the Hawkes loss term maximises the probability of arrival times for each event type, it does not consider the 'competition' for

arrivals across different event types. Based on Daley and Vere-Jones [117], if the arrival time t_i is known, the event type with highest probability of arrival is $\arg\max_k \lambda_k(t_i)$. By minimising the cross-entropy function between the distribution of $\{\lambda_1(t_i), ..., \lambda_K(t_i)\}$ and the real event type x_i, the event type prediction at each timestep can be improved. A similar terminology was also considered in Gu [128]. In this sense, the second loss term can be derived as:

$$L_2 = \sum_{i:t_i \leq T} \sum_{k=0}^{K} -\rho_k(t_i) \log \lambda_k(t_i) \tag{5.8}$$

in which $\rho_k(t_i) = 1$ when $x_i = k$ otherwise $\rho_k(t_i) = 0$.

The third loss term deals with the state transition process by attempting to maximise the likelihood of the coupled state sequence that is sampled from the state transition matrix $\phi_{K \times W \times W}$, in which K and W indicate number of types of events and states respectively, as:

$$L_3 = \sum_{i:t_i \leq T} \log \phi(x_{i+1}, s_i, s_{i+1}) \tag{5.9}$$

The overall loss function combines the aforementioned three losses together: L_1 that maximises the sum of log probability of $p(t)$ in terms of event arrival times; L_2 that maximises the probability that the correct type of event happens at each arrival time; and L_3 that maximises the likelihood of the sampled state sequence. The overall loss is calculated as:

$$L = -L_1 + \eta_1 L_2 - \eta_2 L_3 \tag{5.10}$$

in which η_1 and η_2 are weighting coefficients. The training of parameters can be achieved by applying the gradient descending algorithm. In practice, the loss term L_3 can be separated from the total loss and be trained separately as the parameters involved are isolated from the Hawkes loss term L_1 and cross-entropy term L_2, as conducted in Morariu-Patrichi and Pakkanen [131].

Once the sd-PNHP model is trained, the model is able to predict the time and type of next event arrival. Given the density function for next event time, shown in Eq. 3.25, the expected time with minimum L2 loss can be derived as $\hat{t}_i = \int_{t_{i-1}}^{\infty} t * p_i(t) dt$; and given time t_i, the most likely event type would be $\arg\max_k \lambda_k(t_i)/\lambda(t_i)$; or without knowledge of time, the most likely event type would be $\arg\max_k \int_{t_{i-1}}^{\infty} \lambda_k(t)/\lambda(t) * p_i(t) dt$ [16].

5.3.4 LOB Event Stream Simulation

Given a well-trained model, an event-state sequence can be sampled using Ogata's thinning algorithm [130], shown in Algorithm 4. By iteratively incorporating simulated event and state into the input and removing the first event and state in the input to form new inputs, event and

Algorithm 4 Ogata's thinning algorithm for iterative sampling of event-state stream

Input: $(e_{i-I},...,e_{i-1})$, $(s_{i-I},...,s_{i-1})$
set the upper bound λ^* for all $\lambda(t)$.
for $k = 0$ to K **do**
  let $t^k \leftarrow t_{i-1}$.
  repeat
    sample $\delta t \sim \text{Exp}(\lambda^*)$, $u \sim \text{Unif}(0,1)$.
    $t^k \leftarrow t^k + \delta t$.
  until $\lambda_k(t^k)/\lambda^* \geq u$
end for
$t_i \leftarrow \min_k t^k$, $x_i \leftarrow \arg\min_k t^k$, $e_i \leftarrow (t_i, x_i)$
sample $s_i \sim \phi(x_i, s_{i-1})$
return e_i, s_i

state pairs can be constantly generated. This approach contrasts with the LOBRM simulation presented in the Chapter 4 in the sense that the simulation generated by LOBRM is composed of separate predictions on LOB states, while the intended simulation here is generated as a result of continuous events' interaction and accumulation.

With a model being able to stochastically sample event streams based on its knowledge learned from historical event stream data, a realistic simulation of the LOB data can be achieved. It is worth noting that such a simulation is remarkably different from the mainstream agent-based LOB simulation. The simulation is *realistic* judging from the fact that the generation of events is from knowledge learned on real data instead of from agents with subjectively set strategies and parameters. Such a realistic simulation of the LOB data allows for an alternative way to validate some empirical findings on the real LOB market. To complete the design of the simulator, apart from the type and time of each event, price and volume information affiliated with the event is also needed. Based on empirical studies, the relative price ($p^a - p^{a(1)} \geq 0$ or $p^{b(1)} - p^b \geq 0$) and volume of submitted orders follow a power law distribution [25]. Thus, a truncated discrete power law distribution with parameter a and boundary b in the form of

$$p(x) = 1/(x+1)^a, \quad x = 0,\ldots,b \tag{5.11}$$

is used to fit those distributions from real data. Some special settings concerning order type and price include: the order cancellation on a specific price level follows first-in-first-out terminology; market orders appear as a given percentage of submitted orders; orders with price that cross the spread ($p^a - p^{a(1)} < 0$ or $p^{b(1)} - p^b < 0$) appear as a given percentage of submitted limit orders; the distribution parameters for price and volume is different when bid-ask spread is one or two. With these settings, a minimal and realistic simulation of the LOB can be achieved.

5.4 Experiments

5.4.1 Data Preprocessing

Experiments are conducted on the LOBSTER dataset, providing real world LOB event stream data of five days' length for three stocks, ticker symbol INTC (*Intel*), MSFT (*Microsoft*), and JPM (*J.P. Morgan*). The dataset contains only top five price levels information on both sides of the book. On average there are 0.45 million event updates per trading day per stock. The rolling window approach of step size one is used to form times series samples with length 50.

Table 5.1: Model comparison results for MSFT. Best performance values underlined.

Model	NLL	Time loss	Acc. type
Hawkes	0.68	1.27	42.45 / 43.12
LSTM	-	-	47.72
SAHP	-0.63	1.09	48.53 / 47.77
CT-LSTM	-0.80	1.02	46.29 / 45.89
PCT-LSTM	-0.92	0.99	50.09 / 49.31
PCT-LSTM [non-sd]	-0.91	1.00	49.59 / 48.78
PCT-LSTM [$\eta_1 = 1$]	-0.60	1.03	49.12 / 47.96

Table 5.2: Model comparison results for INTC. Best performance values underlined.

Model	NLL	Time loss	Acc. type
Hawkes	0.80	1.25	39.95 / 39.28
LSTM	-	-	45.97
SAHP	-0.36	1.03	47.11 / 45.85
CT-LSTM	-0.52	0.99	44.26 / 43.20
PCT-LSTM	-0.63	0.95	48.66 / 46.94
PCT-LSTM [non-sd]	-0.62	0.95	48.42 / 46.51
PCT-LSTM [$\eta_1 = 1$]	-0.35	1.01	48.20 / 46.09

Table 5.3: Model comparison results for JPM. Best performance values underlined.

Model	NLL	Time loss	Acc. type
Hawkes	-0.42	1.08	48.13 / 48.11
LSTM	-	-	57.86
SAHP	-1.74	1.01	59.45 / 56.84
CT-LSTM	-2.03	0.90	59.51 / 58.35
PCT-LSTM	-2.14	0.85	61.22 / 59.74
PCT-LSTM [non-sd]	-2.13	0.86	60.77 / 59.08
PCT-LSTM [$\eta_1 = 1$]	-1.66	0.90	60.37 / 54.57

5.4.2 Model Comparison

This section compares testing results for the proposed model against different state-of-the-art models that predict event arrivals. The performance of seven models are compared: (1) Hawkes: stochastic Hawkes point process model with exponential decaying kernel [131]; (2) LSTM: naïve LSTM model, with no modelling of event intensity rates [83]; (3) SAHP: self-attentive neural Hawkes process that uses the attention mechanism instead of recurrent input structure [126, 128]; (4) CT-LSTM: neural Hawkes process that uses a single CT-LSTM unit for the modelling of the intensity rates for all types of events [16]; (5) PCT-LSTM: the proposed state-dependent neural Hawkes process with stacked CT-LSTM units to model the intensity rates for different types of events separately; (6) PCT-LSTM [non-sd]: a variant of the PCT-LSTM model that does not contain an event-state interaction mechanism; (7) PCT-LSTM [$\eta_1 = 1$]: a full PCT-LSTM model trained with a mixed loss function that contains a cross-entropy term. The same metrics used in Mei and Eisner [16] are adopted to measure performance: (1) negative logarithm likelihood (NLL) $p(t)$ per event; (2) next event time prediction accuracy, measured as the absolute distance between real and predicted time after common logarithm operation, which is necessary as the ground truth for time spans from several microseconds to seconds; (3) next event type prediction accuracy in percentage, when the next even time is known and not known. To enable parallel comparison, both event and state are included as inputs for all models, except for the non-state-dependent model.

Model Specifications In terms of parameter settings for all models, all linear layers in LSTM units are set with two layers, 16 units and Tanh activation, except for that the Softplus activation is used for the linear layers for deriving decay coefficient (>0). The attention module in SAHP is of 4 heads. The embedding layers for event type and state indicator are also of two layers, 16 units and Tanh activation. The decoding layer for intensity rates is of one layer, 16 units and Softplus activation that forces outputs to be positive. Optimiser is RMSprop, learning rate is 2e-3 and all models are trained 200 iterations. All experiments are repeated five times to get the mean of criteria.

Testing results are shown in Table 5.1, 5.2 and 5.3. It is found that the performance of neural network family models is better than the pure stochastic Hawkes model in terms of all criteria, as a stochastic Hawkes model is controlled by only a few parameters and the form of the kernel is pre-defined, bounding the expressiveness of the model. This finding is in accordance with existing literature [16, 126]. The LSTM model does not contain a point process terminology to derive the intensity rate, thus only the event type prediction is illustrated here. The performance improvements brought by SAHP or CT-LSTM are mixed, which differs from the conclusion drawn in Zhang et al. [126] that SAHP is better than CT-LSTM. The main reason for this contradiction might be the time series input in this research is comparatively short with a length of 50, whilst the time series studied in Zhang et al. [126] has a maximum length of over 200, and the

self-attention module is more effective for long sequences. PCT-LSTM achieves the overall best performance compared with all alternatives, with the lowest negative log likelihood and time loss, and the highest event type prediction accuracy. This improvement validates that the separate modelling for intensity rates for different event types through the PCT-LSTM unit is effective in multi-variate point process modelling.

The learned state transition matrix is partially shown in Eq. 5.12:

$$\left(\begin{pmatrix} 0.93 & 0.07 & <0.01 \\ \approx 0.01 & 0.99 & \approx 0.01 \\ <0.01 & 0.02 & 0.97 \end{pmatrix}, \begin{pmatrix} 0.97 & 0.03 & <0.01 \\ \approx 0.01 & 0.99 & \approx 0.01 \\ <0.01 & 0.07 & 0.93 \end{pmatrix} \quad ...\right) \tag{5.12}$$

It can be seen that inertia exists in market condition transition, with more than 90% possibility that the market condition remains the same as the condition at last timestep. Distributions of state transition shift from $\phi[0]$ to $\phi[1]$ when the arrived event type changes from a bid submission to an ask submission. This indicates the existence of a feedback loop, in which the current state influences the arrival of next event type and conversely the arrived event type influences the distribution of state transitions. Also, a certain degree of asymmetry exists in state transitions: when arrived event type is a bid submission, the probability of state transforming from an *ask market* to a *neutral market* (0.07) is higher than transforming from a *bid market* to *neutral market* (0.02); yet, when arrived event type is an ask submission, the probability of transforming from an *ask market* to *neutral market* (0.02) is lower than transforming from a *bid market* to a *neutral market* (0.07). This asymmetry is intuitive.

5.4.3 Discussions on the Model Settings

The superiority of stacking CT-LSTM units has been verified in the last section. In this section, the performance improvement brought by the inclusion of a state indicator, and the impact of the modified cross-entropy loss function are discussed. Results are presented in the bottom two rows of Table 5.1, 5.2 and 5.3. Comparing the results from PCT-LSTM [non-sd] with the results from PCT-LSTM, it is found that all criteria are inferior when state information is removed from input, indicating that the market condition indeed helps the modelling of intensity rate evolution and therefore improves prediction. The performance of PCT-LSTM [$\eta_1 = 1$] is then compared with PCT-LSTM. It is found that the inclusion of the cross-entropy loss term does not help improve prediction performance of the PCT-LSTM. However, it is found that this cross-entropy term indeed facilitates prediction for SAHP across all criteria, as indicated in Gu [128], and the testing results of SAHP are obtained by including this term. In this sense, the choice of whether to include the cross entropy loss term can be flexibly determined according to which type of model to be implemented.

To summarize, when compared with all chosen baselines, the proposed PCT-LSTM model achieved the best performance considering all criteria. The reasons for this superiority are

threefold. First, PCT-LSTM uses continuous RNNs to model the evolution of intensity rates. The parameters involved in continuous RNNs are overwhelmingly larger than those in a stochastic Hawkes process, enabling the model a higher degree of expressiveness and flexibility in modeling the influence of past events on the evolution of intensity rates. Second, PCT-LSTM adopts a stacked structure of CT-LSTM units to introduce a fine-grained depiction of the interaction between events, which showed a higher goodness-of-fit compared with the original neural point process model, wherein a single CT-LSTM unit is used to model the intensity rates of different kinds of events. Finally, the proposed PCT-LSTM model is coupled with an event-state interactive mechanism, specifically designed for scenarios in which event arrivals tend to be under the impact of current market conditions indicated by prior stochastic studies [131]. With the help of this interactive mechanism, the performance of PCT-LSTM outperformed all baselines in the end.

5.4.4 Sensitivity Analysis

Here, a minimal simulation of the LOB is attempted by stochastically sampling event stream using Algorithm 4 from the sd-PNHP model, combining with the empirical distributions for order price and volume learnt from the same dataset. Given a random LOB initialisation, LOB simulation can be achieved by accumulating the sampled event stream on the LOB. The simulation is carried out in the ABIDES platform, of which the implementation details are similar to the ones to be presented in Chapter 6. As a starting point, a sensitivity analysis is conducted on a system in which all order streams are generated from the sd-PNHP. Key parameters that relate to the empirical distributions of order price and volume, the percentage of market orders in all orders, and some control parameters that enforce the book will not run out of liquidity, are considered. One thing to be noticed is that neural parameters that dominate the order stream pattern are not involved in sensitivity analysis, owing to its feature of being a 'black box'. The exact parameters involved are listed in Table 5.4, and they are: (1) P: the exponent of power law distribution for order price; (2) $V1$: the exponent of power law distribution for order volume at the top price level (quote prices); (3) $V2$: the exponent of power law distribution for order volume at deep price levels (prices inferior to quote prices); (4) Mi: the market order imbalance index (0 indicating balanced, and $\pm 100\%$ indicating all market orders are bid or ask orders); (5) Mv: the exponent of power law distribution for market order volume; (6) Lb: the lower bound value for volumes at each price level. Once volumes fall below this value, upcoming limit order volumes will be forced to increase 1000 to ensure adequate liquidity in the market; (7) Ip: the possibility for a limit order to fall within the spread (being one tick higher than the best bid or one tick lower than the best ask) when the spread is larger than one. Spread is denoted by sp.

The Sobol's global sensitivity analysis is implemented, as performed in McGroarty et al. [10]. In this variance-based analysis, the ANOVA representation of a function $f(x)$ of $x = (x_1, ..., x_n)$ is:

$$f(x) = f_0 + \sum_{s=1}^{n} \sum_{i_1 < ... < i_s} f_{i_i ... i_s}(x_{i_1}, ..., x_{i_s}) \tag{5.13}$$

Table 5.4: Parameters for the sensitivity analysis on the sd-PNHP based LOB simulation

Parameter	Initial value ($sp = 1$ / $sp > 1$)	Fluctuation range
P - Price dist.	1.5 / 4.7	± 0.25
$V1$ - Volume dist. (top)	0.9 ~1.2	± 0.25
$V2$ - Volume dist. (deep)	0 ~1.8	± 0.25
Mi - Market imbalance	0	± 100%
Mv - Market vol. dist.	1.2 / 1.6	± 0.25
Lb - Lower bound	12500	± 2500
Ip - Inner spread prob.	0.05	± 0.025

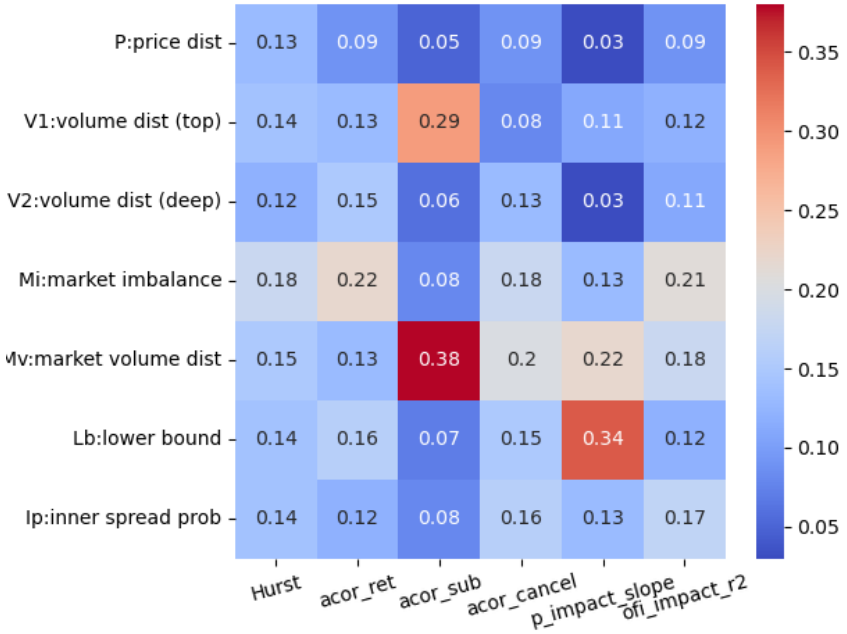


Figure 5.2: Heatmap showing results of sensitivity analysis on the simulated LOB data generated by sd-PNHP. Numbers shown are sensitivity indices standardised column-wise.

in which the s-th summation denotes the s-th order cooperative effect of variables $x_{i_1},...,x_{i_s}$ on the function output. Thus, the total variance and partial variance of $f(x)$ can be denoted as:

$$D = \int (f(x) - f_0)^2 \rho(x) dx \tag{5.14}$$

$$D_{i_1...i_s} = \int f^2_{i_1...i_s} \rho(x_{i_1},...,x_{i_s}) dx_{i_1},...,x_{i_s} \tag{5.15}$$

The total variance concerning variable x_i and its total sensitivity index can be calculated as:

$$D_i^{tot} = \sum_{s=1}^{n} \sum_{\langle i \rangle} D_{i_1 \ldots i_s} \tag{5.16}$$

$$S_i^{tot} = \frac{D_i^{tot}}{D} \tag{5.17}$$

in which the symbol $\langle i \rangle$ sums over all D terms that contain i. Sobol [132] provides an efficient Monte-Carlo method for calculating the indices. According to the method, x is uniformly sampled from the input space. Random perturbations are added to each input element of x, and the perturbed x are used to simulate $\dim(x)+1$ copies of LOBs. Each LOB is equivalent to one hour's length. In total, $100 * (\dim(x)+1)$ LOB samples are generated. Selected stochastic properties of the LOB are used to evaluate the sensitivity of the system, as in McGroarty et al. [10]: (1) the Hurst exponent of volatility [35, 36]; (2) the autocorrelation of mid-price return [133]; (3) the first lag autocorrelation of order-sign for order submissions [35]; (4) the first lag autocorrelation of order-sign for order cancellations [35]; (5) the best exponent β of the price impact function [50]; and (6) the R-squared for the order flow imbalance function [57]. The first four criteria relate to the memory of order flow or the resulting mid-price series, and the last two criteria relate to the market price formation mechanism. Fig. 5.2 presents a heatmap after standardisation of the sensitivity indices.

From Fig. 5.2, it can be seen that the top two parameters that have influence over the stochastic properties considered are: (1) the exponent of distribution for market order volume; and (2) market order percentage imbalance. In McGroarty et al. [10], it was argued that the upper limit of market order volume distribution is the most influential parameter, and the stochastic properties that it affects most are the Hurst exponent and the exponent for price impact function. Both the exponent for market order volume distribution and the upper limit for market order volume directly affect the size of market orders. Thus, conclusions in McGroarty et al. [10] concur with the findings of the sensitivity analysis, even though the exact methodologies adopted differ.

The sensitivity analysis here provides an initial glimpse into the dynamics of the simulated LOB generated by the sd-PNHP model. In Chapter 6, the model will be fused into an agent-based platform to conduct further simulation experiments, through which more insightful results will be drawn.

5.5 Conclusions

This chapter presented the state-dependent parallel neural Hawkes process (sd-PNHP) for LOB event stream prediction and simulation. This research is the first to systematically investigate the utilisation of neural Hawkes models in the field of LOB modelling. The main contributions are: (1) introducing an event-state interaction mechanism into a neural Hawkes process that enables improved prediction performance and efficient sampling; (2) improving the structure

of the CT-LSTM module used in the original neural Hawkes process to a PCT-LSTM module with stacked units, in which the evolution of intensity rates for different types of events are governed by separate kernels; (3) extending the task from LOB event prediction to realistic LOB data simulation by combining a well-trained neural Hawkes model for event type and time sampling, and a rule-based generator for order price and volume information generation. Through experiments, it is found that the proposed model outperforms other mainstream point process models in the specific task of LOB event and time prediction. Meanwhile, the well-trained model was successfully used for LOB simulation facilitated by the Ogata's thinning algorithm. A sensitivity analysis was implemented on the simulation system in order to gain an initial understanding of the dynamics of the system, which paves the way for conducting further comprehensive LOB simulation experiments in Chapter 6. This way of generating simulated LOB data provides a new alternative to existing methods. As a whole, this research has the potential to benefit both financial microstructure researchers and high frequency trading practitioners.

Meanwhile, the performance of the sd-PNHP is only validated on the task of LOB event prediction. Future studies can be carried out to test whether the proposed model can also benefit other types of tasks. On the other hand, several settings of the simulator are simplistic, for instance the occurrence of market orders and the relative price distribution of arriving orders. For enhanced realism, more advanced and realistic features should be introduced into the simulator. The proposed sd-PNHP based simulator can also be combined with agent-based simulators to develop a hybrid simulator that enjoys unique characteristics from both methodologies.

CHAPTER 6

NEURAL STOCHASTIC AGENT-BASED LOB SIMULATION

The content to be presented is a generalisation of published [134] and under-review works.

6.1 Introduction

LOB data is of critical importance in financial microstructure studies, and the majority empirical and theoretical studies were carried out on the basis of historical LOB data. Nevertheless, using historical LOB data suffers from two major problems. First, as one cannot interact with historical data, it cannot be used to conduct 'what if' counterfactual analysis. Therefore, backtesting a trading algorithm on historical data suffers from the unrealistic assumption that the market will not react. This is a particularly dangerous assumption when trading large volumes that will move the market. Second, the availability of historical LOB data is limited and expensive. Simulations and synthetically generated data, respectively, can help to overcome these problems.

Two mainstream models for LOB simulation and synthetic data generation are agent-based models (ABMs) and stochastic models (SMs). ABMs construct an interactive trading environment containing heterogeneous agents with behavioural logic that is human-defined. While defined agents are often modeled as simplifications of real world actors such as market makers, momentum traders, and liquidity providers, the objectivity of ABMs can be doubted as one cannot guarantee that the simulation will behave in the same manner as the real world [3]. On the other hand, SMs learn stochastic properties directly from historical data and so are objectively grounded in reality. Prior studies have shown that market characteristics, such as price and volume distributions and the arrival of market events, can be captured using SMs, and this learned knowledge can be used to generate realistic synthetic data. However, SMs have the limitation of not being interactive. Therefore, while SMs can generate endless amounts of synthetic data, the problems associated with backtesting on real data remain. Others have previously

suggested the advantages of combining models [135], however, there are few prior studies that have attempted to embed SMs of the LOB in an interactive ABM environment. These include Panayi and Peters [136], who claim their model is a hybrid of ABM and SM; however, the model is essentially a stochastic model in which order flows are attributed to imaginary agents, and there is no provision to interact with the stochastic system. More recently, Kumar [137] proposed an ABM model in which a stochastic market maker is embedded and the market maker interacts with other trading agents. However, the focus of that study is on deploying the stochastic model as a strategic trader to make profit; the model is not designed as a realistic market simulator to investigate market dynamics. The most similar research that we have found is Coletta et al. [111], in which a conditional generative adversarial network (CGAN) based agent is first trained on real data, and is then used as a *world agent*. The world agent is then incorporated into an ABM simulation to conduct interaction experiments. Nevertheless, CGAN agent lacks support from a statistical perspective, being a complete deep learning 'black box'. Within such a simulation, the variations in model results are hard to be explained, thus hamper the transparency of the system. Besides, the CGAN model cannot be validated through comparing prediction accuracy with mainstream models, while the state-dependent parallel neural Hawkes process (sd-PNHP) model can be rigorously grounded on real data using criteria like log likelihood of event arrival time and event type prediction accuracy. Thus, we propose that there is a research gap, which we attempt to address in this paper.

This chapter delves into the intricacies of simulating the LOB using a pioneering approach: the neural stochastic agent-based model (NS-ABM). This model emerges as a unique blend of the conventional ABM and SM methodologies. The primary contributions of this research are outlined as follows:

- At the core of the NS-ABM lies the neural stochastic background trader (BT). This component is designed to simulate the collective aggregation of order events characteristic of the entire market. The BT utilises an advanced sd-PNHP model to learn the behavioural pattern of the whole market from historical real world level-2 LOB data. When turned to sampling mode, the BT can iteratively sample the type and arrival time of next event according to knowledge learned. The model uses neural methods to model the intensity rate curves of different types of events, achieving a balance between model expressiveness and interpretability.

- The deep diffusion probabilistic model (DDPM), a non-parametric generative method for learning data distributions, is utilised to learn the order-related attributes (e.g., price and volume) conditioned on various market indicators. As observed on the empirical data, order-related attributes have significant dependency on various market indicators (e.g., bid-ask spread and market imbalance). The DDPM's learned conditional distributions provide auxiliary information for the BT in event sampling.

- The BT that equipped with sd-PNHP and DDPM modules is incorporated into the open-source Agent-Based Interactive Discrete Event Simulation (ABIDES) [138] framework. The BT is shown to produce LOB dynamics that reproduce a comprehensive list of more than ten *stylised facts* about real world LOBs. Such high fidelity has not been shown in previous LOB simulation studies. Concentration on the interpretability of the results produced by the BT has also been made, by looking deep into the mechanism producing those results.

In the experiment section, the *stylised facts* reproduced by the simulation when the BT is running on its own is evaluated and discussed. Further, a population of trading agents with various *trend* and *value* trading strategies are introduced into the simulation. It demonstrates that the BT reacts realistically to endogenous events caused by other trading agents. The resultant LOB dynamics continue to exhibit the *stylised facts* of real markets, which demonstrates the realistic behaviour of the ABM. It is also shown that the simulation exhibits order flow impact and financial herding behaviours that are similar to empirical observations of real markets. Lastly, an explicit investigation on the price impact in the system is presented, discussing how it mimics and deviates from empirical studies.

6.2 Model Formulation

The NS-ABM for LOB simulation encompasses two primary components. Firstly, the simulation employs an agent-based approach, utilising the ABIDES open-source LOB simulation platform [138]. Secondly, a neural stochastic BT is trained, with its behavioral logic derived from the sd-PNHP model and the DDPM model based on historical trading data. This BT is then integrated as an agent into ABIDES, accompanied by the introduction of other trading agents that can engage with the BT. A graphical illustration of the proposed NS-ABM framework is shown in Fig. 6.1.

6.2.1 Agent-Based Interactive Discrete Event Simulation

ABIDES [138] is an agent-based simulation framework used to generate high-fidelity LOB data and conduct microstructure experiments. It has been adopted in varies studies, e.g., for developing and evaluating trading agents [139], and for investigating market manipulation [140]. ABIDES mimics real market settings in several ways: (1) it has a realistic messaging system derived from NASDAQ's published equity trading protocols, ITCH and OUCH; (2) it has no assumptions or restrictions on market settings, such as order size or time intervals between discrete events; and (3) it is equipped with a set of classes and functions that enable extension of existing agent types and actions. Fig. 6.2 is a simplified version of the simulation logic shown in Byrd et al. [138]. In the following paragraphs, two general classes of ABIDES' high frequency trading (HFT) agents used in this paper are introduced: *trend* trading agents and *value* trading agents.

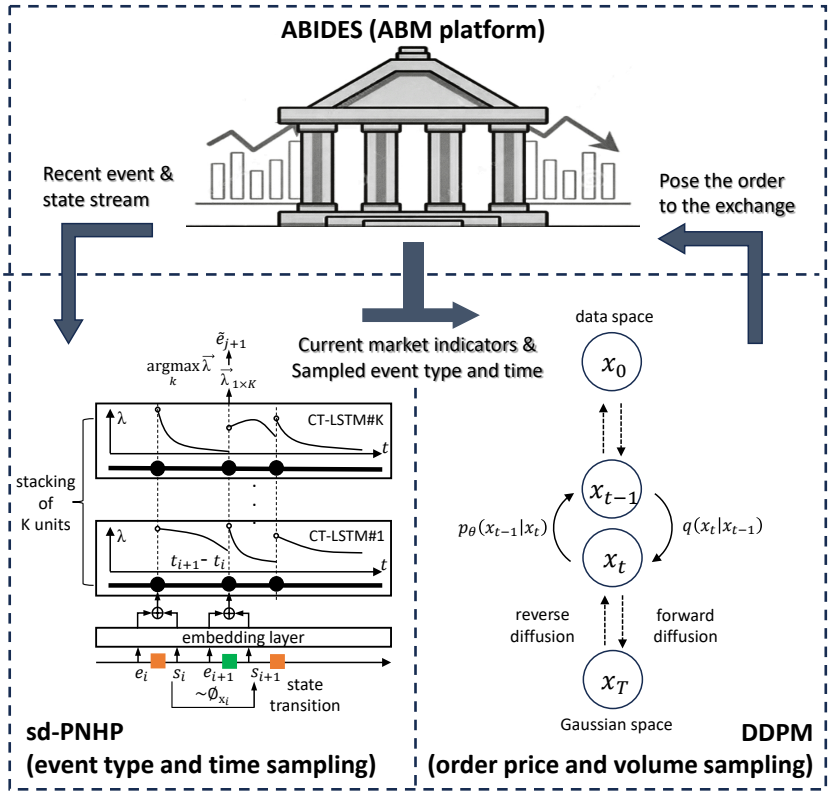


Figure 6.1: The framework for NS-ABM. The BT is composed of an sd-PNHP module for event type and arrival time sampling, and a conditional DDPM module for order price and volume sampling. Explicitly, the sd-PNHP is subscribed to market updates, receiving recent event and market state streams from ABIDES. The sd-PNHP then derives the intensity rates for four different types of events, on which Ogata's thinning algorithm is used to sample next event type and arrival time. The sampled event, together with information concerning current market indicators (e.g. bid-ask spread and market imbalance), are then passed to the DDPM module as conditional information. The DDPM first samples from Gaussian space, and gradually removes noise based on the conditional information. When the maximum denoising step is reached, the sample concerning order price and volume is drawn. Finally, the order is submitted to ABIDES exchange LOB and the BT waits for the next market update.

Trend agents assume that a trend in price will persist or reverse in the near future, and they are called momentum (MM) agents and mean reversion (MR) agents, respectively. *trend* agents calculate the current trend in price using the difference between moving average price over a long time window and over a short time window. The trading logic of *trend* agents is illustrated in Algorithm 5. These agents try to maintain a neutral position, so the submitted order size is either equal to the size that can close the current position, otherwise a standard size u is used. For instance, assume an agent has a short stock holding of -200. If the agent then decides to submit a market buy order, the order size would be 200 to close the position; conversely, if the agent decides to submit a market sell order, the order size would be u to further expand the

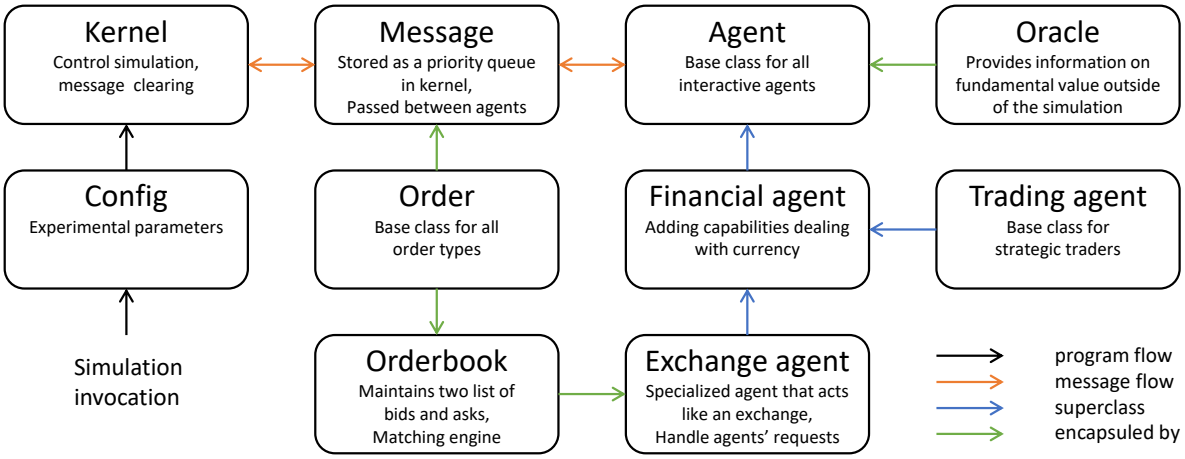


Figure 6.2: Simulation workflow of ABIDES, adapted from Byrd et al. [138]. The simulation starts from defining a *config* file, specifying the number and parameters of agents holding different trading strategies, together with parameters concerning the asset to be traded. The *kernel* is in charge of governing the simulation, passing and clearing message between the exchange and the agents. The *message* base class is the intermediate through which exchange of information happens, for instance 'wakeup calls' that wake up an agent to act, and 'limit order submission' that indicates the intention of order submission from the agents to the exchange. The *agent* base class is what strategic traders are defined upon, being able to process information from market updates and submitting orders to, or cancelling orders from, the exchange. Certain types of agents rely on an external valuation *oracle* to make decisions. The *order* base class is what market orders and limit orders are defined upon, and all orders are passed to the *order book* before either resting on the book or being immediately transacted.

position. By maintaining a neutral position, *trend* agents attempt to profit from high-frequency trading; rather than profit from capital gains by holding a large position over a comparatively long time horizon.

Value agents trade according to observations on a fundamental value oracle that is exogenous to the market [141]. Such an oracle can be any time series generated through a stochastic process. The SparseMeanRevertingOracle class is used to generate fundamental values for *value* agents. As presented in Byrd [141], the fundamental value follows an Ornstein-Uhlenbeck process, of which the value at t_i is denoted as:

$$p_{t_i} = \mu + (p_{t_{i-1}} - \mu)e^{-\gamma\delta_i} + u_{t_i} \tag{6.1}$$

$$u_{t_i} \sim N(0, \frac{\sigma^2}{2\gamma}(1 - e^{-2\gamma\delta_i})) \tag{6.2}$$

in which δ_i denotes units of time elapsed since last observation. When δ_i increases, the fundamental value and its variance at t_i converge to μ and $\frac{\sigma^2}{2\gamma}$, indicating its mean-reverting essence.

Before agents observing the fundamental value, they first update internal estimates $\tilde{p}_{t_i}$ and

Algorithm 5 Trading logic for *trend* agents (MM & MR)

Require: window size l_1 and l_2, $l_2 > l_1$, current stock holding h

if receive wake up call from the kernel **then**
  cancel all unexecuted orders
  send request for quote prices to the exchange agent
  receive quote prices $(p_t^{a(1)}, p_t^{b(1)})$ from the exchange agent
  calculate mid price p_t^{mid} and append it to the agent's internally stored list mid_list
  if len(mid_list) >= l_2 **then**
    if $\text{MA}_{l_1}(mid\_list) > \text{MA}_{l_2}(mid\_list)$ **then**
      if type(agent) == MM **then**
        submit a market buy order, $q = \max(-h, u)$
      else if type(agent) == MR **then**
        submit a market sell order, $q = \max(h, u)$
      end if
    else
      do steps 8-11 using inverse logic (i.e., MM sells, MR buys)
    end if
  else
    queue next wake up call in kernel, break
  end if
  queue next wake up call in kernel
end if

$\tilde{\sigma_{t_i}}^2$:

$$\tilde{p}_{t_i} = (1-(1-\gamma)^{\delta_i})\mu + (1-\gamma)^{\delta_i}\tilde{p}_{t_{i-1}} \tag{6.3}$$

$$\tilde{\sigma}^2_{t_i} = \frac{1-(1-\gamma)^{2\delta_i}}{1-(1-\gamma)^2}\sigma^2 + (1-\gamma)^{2\delta_i}\tilde{\sigma}^2_{t_{i-1}} \tag{6.4}$$

After observing the fundamental value as o_{t_i} with noise σ_o^2, estimates are updated in a Bayesian manner:

$$\tilde{p}_{t_i} \leftarrow \frac{\sigma_o^2}{\sigma_o^2 + \tilde{\sigma}^2_{t_i}}\tilde{p}_{t_i} + \frac{\tilde{\sigma}^2_{t_i}}{\sigma_o^2 + \tilde{\sigma}^2_{t_i}}o_{t_i} \tag{6.5}$$

$$\tilde{\sigma}^2_{t_i} \leftarrow \frac{\tilde{\sigma}^2_{t_i}\sigma_o^2}{\tilde{\sigma}^2_{t_i} + \sigma_o^2} \tag{6.6}$$

Upon finishing updating its interval values, the agent makes prediction on the fundamental value at $t_i + \Delta t$ as:

$$\tilde{p}_{t_i+\Delta t} = (1-(1-\gamma)^{\Delta t})\mu + (1-\gamma)^{\Delta t}\tilde{p}_{t_i} \tag{6.7}$$

of which is the value that *value* agents used to compare with the realised stock price at t_i to make trading decision. Here the agents focus on the profit over short terms (in Δt), the same as momentum traders. The estimated fundamental value represents the 'fair price' of the stock to the agent. If the current price on the LOB is underestimated, the agent tends to buy; otherwise

Algorithm 6 Trading logic for *value* agents (ZI & HBL)

Require: Intended maximum surplus r_{max}, look back period l, current stock holding h

if receive wake up call from the kernel **then**
  cancel all unexecuted orders
  send request for quote prices to the exchange agent
  receive quote prices $(p_t^{a(1)}, p_t^{b(1)})$ from the exchange agent
  obtain a noisy observation from the oracle, and update internal estimation on current value p_t in a Bayesian manner
  make estimation on future fundamental value $p_{t+\Delta t}$
  sample requested surplus $r \sim \text{Unif}(0, r_{max})$
  if rand$(0,1) > 0.5$ **then**
    submit a limit buy order, $q = \max(-h, u)$
    if type(agent) == ZI **then**
      if $p_{t+\Delta t} - p_t^{a(1)} > r$ **then**
        $p = p_t^{a(1)}$
      else
        $p = p_{t+\Delta t} - r$
      end if
    else if type(agent) == HBL **then**
      calculate execution probability vector $Prob$ at all price levels $P = (p_{min}, ..., p_{max})$ during past l transactions
      calculate expected surplus $S = Prob * (p_{t+\Delta t} - P)$
      $p = p_{min} + \text{argmax} S$
    end if
  else
    do steps 9-19 using inverse logic (i.e., submit a limit sell)
  end if
  queue next wake up call in kernel
end if

the agent tends to sell. One type of *value* agent makes decisions based entirely on the observed fundamental value, and these agents are termed *zero intelligence* (ZI) agents; the other type of *value* agent use additional information from the LOB to improve the possibility of order execution, and these agents are termed *heuristic belief learning* (HBL) agents. The private value setting over holding preference is intentionally removed. As a result, the agents do not have preference over a long or a short position, to force agents to solely focus on profitability. That is, when there is no price information the agent has no preference over holding 100 units or -100 units of stock. As *trend* agents do not have the private value setting in the ABIDES implementation, the removal of this setting also allows better comparison. The trading logic of *value* agents is described in Algorithm 6.

Trend agent and *value* agent types are representative of common trading strategies and have been widely studied in both analytical models and empirical studies. To aid analysis, a minimal set of trading strategies is deliberately chosen, therefore these are the only two types of strategic trading agents that are considered in this work. However, it is trivial to include other pre-defined agent types, or define new trading strategies within ABIDES. Such explorations of more complex markets are reserved for future work.

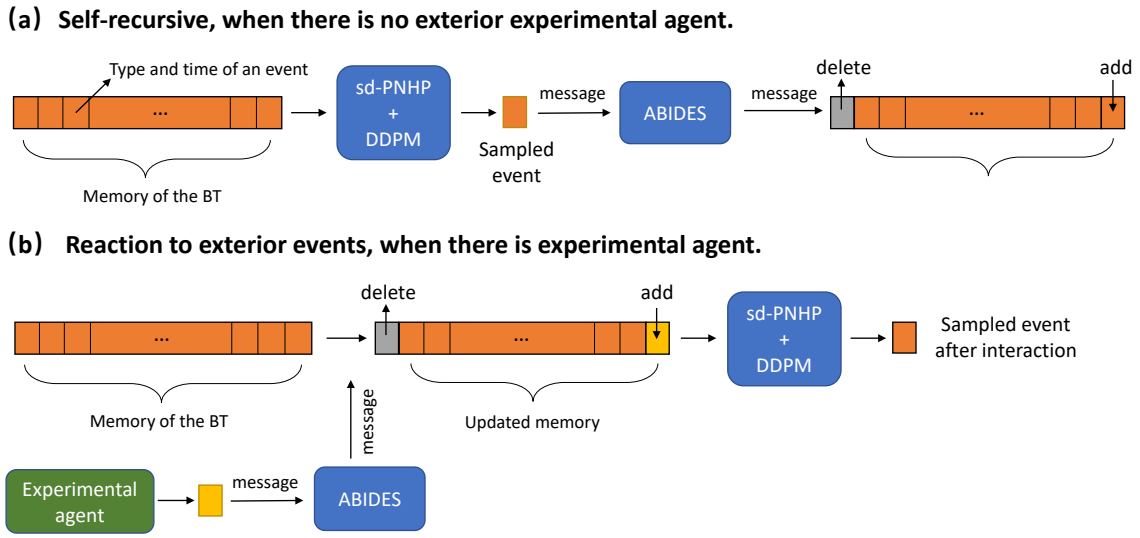


Figure 6.3: The interaction mechanism between the BT and ABIDES: (a) shows how the BT samples events when it is running on its own. A defined length of memory is used as input into the sd-PNHP model, coupled with distributions concerning orders (either empirical or learned through DDPM), to sample a new event. The event is then sent to the exchange, and once accepted will be updated into its memory; (b) shows how interactions with agents take place. It follows a similar logic of (a), however, here the BT will first combine the experimental agent's event into its memory, and then sample events. The sampled event is therefore influenced by other agents' behaviours.

6.2.2 Neural Stochastic Background Trader

Pure ABMs in which all agents act according to human-defined rules can lack objective grounding. Although the simulation framework can be designed to closely imitate the real world (e.g., in ABIDES the messaging mechanism originates from NASDAQ, the LOB operates using real world rules, and the terminology of agent strategies can be found in real markets), realistic dynamics are not guaranteed. Critically, the behavioural patterns of agents, the parameters that determine those behaviours, and the complex interactions between agents combine to generate the overall order stream pattern. The parameters of individual agent behaviours cannot be learned from the aggregate order stream of real markets, as the actions of individual traders are not known (data is anonymous and actions cannot be linked to individuals). However, it *is* possible to learn the overall order stream pattern of the whole market using a sd-PNHP model. Here, the approach of sd-PNHP is adapted to learn aggregate order streams of real data and then deploy the model as an autonomous agent within an ABM so that it can interact with and adapt to the actions of other trading agents in the market. It is hypothesised that this 'hybrid' simulation approach can generate more realistic market dynamics within which new trading strategies can be more rigorously tested.

6.2.2.1 sd-PNHP Based Background Trader

To this end, a neural stochastic BT is implemented in ABIDES. The agent is backed with a sd-PNHP model that is trained on historical LOB event stream data within top five price levels. The agent can make predictions on next event type (i.e., ask submission, ask cancellation, bid submission and bid cancellation) and next event time based on a defined length of event history. By iteratively incorporating newly sampled events into memory as input, the agent is able to endlessly generate event stream data that closely mimics real market data. The agent is termed a *background trader* because: (1) when the agent is running on its own (i.e., being the only agent in the simulation), the order stream it generates represents the logic of a complete market; (2) when the agent interacts with other trading agents (i.e., being part of a multi-agent simulation), the reactions it produces are similar to aggregate market responses. Fig. 6.3 illustrates the logic of the BT when it is running on its own, and when it is interacting with other agents. By defining the BT as a subclass of the trading agent class in ABIDES, it inherits full functionality of communicating with the exchange, submitting and cancelling orders, and keeping track of its real-time profitability.

After the model is well-trained on historical data, a sampling method like the Ogata's thinning algorithm can be used to sample the most likely next event type and arrival time. Experiments performed in Chapter 5 demonstrated that: (1) the model can make predictions with 50%-60% accuracy in event type prediction (compared with baseline accuracy of 25% for this four class classification problem, with classes: ask submission, ask cancellation, bid submission, and bid cancellation); and (2) the model can be used to iteratively sample event streams, either on a self-recursive way or being interactive with other experimental agents. When combined with distributions of order price and order volume, it can be used to generate high-fidelity LOB data that exhibits multiple *stylised facts* found in real data, which is illustrated in the experiment section.

6.2.2.2 Rationale for Choosing sd-PNHP over CGAN

Recent work using a CGAN framework has had success in learning market event patterns [111]. Here, neural point process (NPP) models and generative adversarial network (GAN) models are compared to provide a rationale of why the sd-PNHP approach is chosen, rather than the CGAN approach in Coletta et al. [111]. The arguments are arranged into three perspectives: modelling approach, interpretability, and model validation.

Modelling Approach NPPs explicitly model the event generation process in the LOB simulation. They capture the temporal dependencies and interactions between events based on stochastic processes, such as Poisson processes or Hawkes processes. These models incorporate domain knowledge about event dynamics and market microstructure, allowing for explicit control and interpretation of the simulation process. In comparison, GANs do

not explicitly model the event generation process. Instead, they learn the underlying data distribution from training data and generate samples by optimising a generator and discriminator network adversarially. Therefore, GANs focus on capturing high-level statistical patterns in the data without directly modelling the specific mechanisms of event generation in the LOB.

Interpretability NPPs offer greater interpretability due to their explicit modelling of event generation mechanisms and interpretable parameters. Researchers can analyse the model equations, parameters, and the underlying stochastic processes to gain insights into the LOB dynamics. This transparency enables a clear understanding of how different factors impact event generation and provides interpretability to the simulation results. In contrast, GANs generally have lower interpretability due to their complex and opaque architecture. The internal workings of GANs, such as the specific patterns learned by the generator and the decision-making process of the discriminator, are often challenging to interpret. Therefore, understanding the reasons behind specific event generation or how different inputs affect the generated data can be difficult.

Model Validation NPPs are typically trained using maximum likelihood estimation, or other optimisation techniques, to fit the model to observed event sequences. These models can be validated against empirical data by comparing the simulated event sequences with real world LOB data, allowing for quantitative assessment of their performance. In comparison, GANs are trained using an adversarial process, where the generator network is optimised to generate data that will fool the discriminator network. Evaluating the performance of GANs is often challenging, as it relies on subjective assessments, visual comparisons, or domain-specific metrics. Therefore, quantitatively validating GAN-generated LOB data against empirical data can be more difficult, making it harder to establish the trustworthiness of the simulation results.

6.2.3 Deep Diffusion Probabilistic Model Based Order Attribute Generator

Apart from order type and arrival time, order price and volume are also essential attributes needed to construct a self-sustaining event-based LOB simulator. In Chapter 5, the parametric power law distributions are used to learn the distributions for order price and volume through maximum likelihood estimation (MLE) on the original (real) data. Under this formulation, during the sampling phase the relative price op is first sampled from the empirical distribution ϕ_c^{op}, in which c indicates condition (i.e., bid-ask spread). The order volume ov is then subsequently sampled from the empirical distribution $\phi_{op,c^{'}}^{ov}$, of which the subscript of $op, c^{'}$ indicates the sampling of order volume is conditioned on the formerly sampled order price and other conditions (i.e., bid-ask spread, and whether the order is a market order or limit order).

This parametric way of sampling with limited conditioning provides a minimal simulation for order related attributes. However, it suffers from several limitations:

Strict Power Law Form The distributions of relative order price and order volume do not always follow a power law. For example, regarding order price, the empirical study by Bouchaud et al. [25] indicated that only non-negative relative prices follow a power law distribution, while the conclusion does not hold for negative relative prices (a negative relative price indicates an order falls within the spread; i.e., within the range between current best bid and current best ask).

Limited Conditioning The conditions on which the distributions are based do not cover a broad range of market variables. Previously, the condition only contains the bid-ask spread. Instead, through observations on the real data, it is found that the distribution shape of order attributions also dynamically changes with conditions like current volume on each price levels, and order imbalance between the two sides of the LOB.

Considering the limitations listed above, the sampling method for order attributes should contain the following desirable features:

Non-Parametric Learning The model should be non-parametric. It should not specify the distribution shape and parameters, and it should have the freedom to learn the distribution form directly from data.

Complex Conditioning The model should be able to model complex conditional relations between variables. These include conditioning between the context and the attributes, and the conditioning between different dimensions of the attributes. The context on which the attributes are conditioned can be both discrete or continuous variables.

Generative Approach The model should be generative rather than deterministic, such that the model learns the distribution of the data and has the ability to efficiently sample data from the learned distribution.

Deep generative models [142] are non-parametric methods that use neural networks to learn a data distribution from high-dimensional data. These models do not pose restrictions on the exact form of the distribution and are capable of incorporating conditional information (either discrete or continuous) and conducting efficient sampling (transferring data sampled from a simple prior, mostly a Gaussian, to the ground truth distribution). Among the family of deep generative models, the DDPM [21] has achieved state-of-the-art performance in conditional generation tasks, with particular success in text-to-image generation [143]. In comparison with other mainstream deep generative models, such as GANs, it has been demonstrated that DDPM shares both the characteristics of high-fidelity sample generation and stable training. The core

idea behind the deep diffusion model is to learn a bi-directional mapping between the data space and the multi-variate Gaussian space. The forward diffusion path with fixed noise scheme transfers the original data into white Gaussian space, while the backward diffusion path with learnable neural parameters enables the transfer of samples from Gaussian space into the original data space.

6.2.3.1 DDPM-Based Order Attribute Generator

Orders have four attributes: *type* and *arrival time*, which are sampled from the sd-PNHP; and *relative price* and *volume*, which are generated by DDPM. DDPMs are commonly used for generating image data. However, unlike image data, which is regularised (i.e., three channel RGB or two channel B&W), order price and volume are essentially multi-modal information that require additional care. Here, the process used to generate order price and volume is described.

For order price, the common practice of considering *relative* price [25] rather than *absolute* price is taken. Relative price is calculated as the difference between the order's price and its affiliated side's current best quote price. As the simulation of a *large-tick* stock [121] (i.e., a highly liquid and low price stock) is intended here, the price difference between different price levels on one side of the book is dominantly one tick, and the bid-ask spread is rarely larger than two ticks. Therefore, in a LOB containing five price levels, the relative price of limit orders falls within the discrete range of $\{-2,\dots,0,\dots,4\}$. For market orders, the representation of price is unified as $-spread$ (i.e., the relative price of the best order on the opposite side of the book). Under this formulation, the value range for order prices is $op \in \{-2,\dots,0,\dots,4\}$, with a lower degree of eagerness (or higher degree of patience) for order execution as the relative price increases [7].

For order volume, orders are empirically found to be mostly placed in units of 100. Thus, all volumes are first divided by 100 and rounded with a minimum value of one. To remove outliers, volume values are further clipped at the 99.9 percentile ov_{max}. As a result, the value range for order volume is $ov \in \{1,2,\dots,ov_{max}\}$. Finally, both price and volume values are standardised into the range $[-1,1]$.

Through empirical observations on the LOB data of several stocks traded on real exchanges, it was found that the distribution of op and ov dynamically changes when some of the market variables fluctuate. For instance, the distribution of op significantly changes when the bid-ask spread enlarges from one to two (as illustrated in Fig. 6.6); and the mean of order volume arriving at a certain price level is related to the unexecuted order volume left on the level. Even though empirical market research indicated that the long term distributions of op and ov remains stable [25], the short term changes in these distributions are essential in replicating LOB patterns that result from HFT. Here, it is assumed that the variation in distribution of op and ov is conditioned on several market variables: (i) bid-ask spread, $sp \in \{1,2\}$; (ii) bid-ask volume imbalance vi, calculated as the difference between the sum of order volume on the bid side and the ask side, divided by the sum of order volume on the LOB; (iii) exact order volume on the $\{1,\dots,5\}$-th level of

the $\{bid, ask\}$ side $v = \{v_1^b, \ldots, v_5^b, v_1^a, \ldots, v_5^a\}$, clipped at the 99.9 percentile lob_{max}; (iv) the sampled event type from sd-PNHP e, encoded as one-hot vectors of size 1×4. All features are standardised into the range of $[0, 1]$. A multi-layer perceptron (MLP) consisting of three linear layers with LeakyReLU activation is used to embed the conditional information, and the output is used as condition in deriving error term $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})$.

Even though DDPM is mainly used in the domain of image generation, it is demonstrated later in the experiment section that DDPM can also be used to model the complex distribution of $\mathbf{x}_0 = (op, ov)$: not only being able to replicate the overall distribution shape, but also able to grasp the dependency between conditional information and $\mathbf{x}_0$.

6.2.4 ABIDES Implementation Details

The ABIDES base class of TradingAgent is extend to implement the BT. The main functionalities of the base class include being able to send and receive messages (including posting order-related actions to the exchange and receiving market information from the exchange), submit and cancel orders, and maintain its own order history. The BT is subscribed to market information and whenever the LOB is updated – either from its own action, or the action of another agent – the BT receives a market subscription message and samples an event (with event type, arrival time, and order attributes). The sampled event will not necessarily to be posted to the exchange, as there is a possibility that an event posted by another experimental agent will arrive earlier. Under this circumstance, the BT will sample a new event and the original event will be abandoned. The newly sampled event is considered an "exterior" interaction (as shown in Fig. 6.3(b)).

In addition, the following functions were introduced to enable BT embedding in ABIDES:

Empty Book: During market open, when there are no orders on the ask (or bid) side, the spread will be assumed to be one tick. If the BT samples an ask (or bid) cancellation action, it will be automatically replaced by an action of submitting bid (or ask) limit order of volume 100 at the top price level. This mechanism is only activated on the rare occasions when one side of the LOB runs out of liquidity.

LOB Liquidity Maintenance: When the LOB runs out of liquidity, the volume of orders arriving at a price level with less than the lower bound units of orders (set as 10^4) is increased by 500. Likewise, when there is too much liquidity on a price level, the volume of orders arriving at a price level with more than the upper bound units of orders (set as empirical 90 percentile: 3.6×10^4) is decreased by 500.

LOB Spread Maintenance: Whenever the bid-ask spread is larger than two ticks, the next order submission will fall within the spread to minimise the spread. This follows the empirical distribution of the bid-ask spread for *large-tick* stocks, which is dominantly no more than two ticks.

LOB Deep-Level Maintenance: As the sd-PNHP is only in charge of sampling events related to the top five price levels of the LOB, the cancellations of orders falling outside the top five price levels (if there are any) happen together with normal sampled cancellations with a pre-defined probability (set as 10%).

Volume Outliers: Outliers (> 99.9%) in the order volume distribution mainly arise from market orders, and the DDPM module does not model outliers. Therefore, the volume outliers in market orders are modelled as ABIDES 'shocks', such that 3% of market orders will become 'shocks' with a mean volume of 1.3×10^4 (both statistics are derived from data).

Market Open: When the market opens, the first five price levels are randomly populated with limit orders of volume $ov \in \{100, 200, \ldots, 1000\}$, until the order volumes on each price level reach a random aggregation value in the range 1.5×10^4 - 2.0×10^4 (derived from the empirical average value of the top five price levels in the dataset). Time intervals for these orders are set as 10^{-6} seconds for fast population of the LOB, during which time no market information will be sent out and no agents will react. These orders can be considered as pre-market-open orders, and the resultant LOB is used as an initial state on which following events accumulate.

6.3 Experiments

6.3.1 Model Learning and Parameter Settings

The sd-PNHP and DDPM models trained on INTC data from the LOBSTER dataset is chosen as the main models to be used in the following experiments. Parameters for experimental agents are human defined, taking reference from the original ABIDES settings.

In sd-PNHP, the majority of linear layers in stacked continuous-time long short-term memory (PCT-LSTM) units are set with two layers, 16 units, and Tanh activation; however, Softplus activation is used for the linear layers for deriving decay coefficient (> 0). The embedding layers for event type and state indicator are also of two layers, 16 units, and Tanh activation. The decoding layer for intensity rates is of one layer, 16 units, and Softplus activation that forces outputs to be positive. Sample length is set as 50, the optimiser used is RMSprop, the learning rate is 2×10^{-3}, and the model is trained for 200 iterations. In DDPM, all linear layers are set with three layers, 32 units, and LeakyReLU activation. The maximum diffusion step is 100, with noise scheme β ranges from 1×10^{-3} to 2×10^{-1}. An exponential moving average parameter updating method is used, the optimiser used is RMSprop, the learning rate is 2×10^{-4}, and model is trained for 200 iterations.

6.3.2 Model Interpretability and Transparency

One of the biggest advantages of the proposed LOB simulation model lies in its enhanced interpretability, when compared with alternative 'black box' deep learning based simulation models. While deep learning models are utilised as a foundation for the proposed simulation, they primarily serve as auxiliary tools to support the stochastic modelling of the LOB; a concept that has been extensively validated in the existing literature. Under this approach, simulation results are more accessible to logical interpretation, which is critical for building trustworthy systems with real world application. Here, elaborations on model interpretability are made from the perspectives of the sd-PNHP for order type and arrival time generation, and the DDPM for order price and volume generation.

6.3.2.1 Order Type and Arrival Time Generation

The sd-PNHP model is in essence a stochastic point process model that uses intensity rate curves to depict the arrival of events, with the help of PCT-LSTM to enhance its expression capability. Therefore, a change in simulation results can be linked directly to a change in intensity rate curves, which can be further traced back to a change in input variables such as the market state or the most recent event history. This help us to identify some of the key causal relations in this complex system.

Here, 10,000 repeated trials of non-iterative single-step sampling are performed. Experiments and results are described below:

- **The influence of market state indicator on the imbalance in upcoming order type**. This influence is tested by setting three parallel sets of experiments, in which the market state indicators are set equal to either 0, 1 or 2 at all timesteps. Other variables are randomly sampled and remain equal across different experiments. The state indicator's influence on the upcoming order flow imbalance is investigated, calculated as the difference between net (submission minus cancellation) bid orders and net ask orders. It is found that when market indicator increases (the volume at the bid quote price grows relatively larger than the volume at the ask quote price), the number of net bid orders gradually overpasses net ask orders. In summary, this phenomenon indicates that the sd-PNHP model tends to grasp the market inertia in the sense that a market with higher aggregate ask (or bid) volume tends to attract a higher net ask (or bid) order inflow.

- **The influence of recent event stream on event type sampling**. This influence is tested by setting four parallel sets of experiments, where all event types in each experiment are set as either 0, 1, 2, 3. Other variables are randomly sampled and remain equal across different experiments. The test sums up event types being sampled in each set of four experiments and discover the top two most likely event types to be sampled after a burst of homogeneous event stream. Through experiments It is found that the submission of

orders tends to be followed by order cancellations from both the ask and bid side, while the cancellation of orders tends to be followed by order submission or cancellation on the same side of the LOB. This phenomenon reveals the symmetrical event interaction in a broad sense. It also indicates that submission of orders on one side the LOB has a higher potential to spread its influence to the other side of the LOB, which might be partially owing to the direct impact laid by some orders (e.g., market orders) on the liquidity resting on the opposite side of the LOB.

- **The influence of historical inter-arrival time on the arrival time of the next event**. This influence is tested by setting four parallel sets of experiments, where the time intervals in each subsequent set are ten times longer than those in the previous set (i.e., the time intervals in the fourth set are 10^3 times longer than the first set). It is found that the influence of increase in historical inter-arrival times on the increase in the time interval until next event arrival is concave. Moreover, when the sd-PNHP is set to run iteratively, the arrival of events will tend to converge back to a comparatively high frequency even after a period of less frequent trading activities, which reveals the high-frequency essence of this LOB simulator.

6.3.2.2 Order Price and Volume Generation

The DDPM learns the bi-directional non-parametric mapping between the data distribution and Gaussian space. By sampling from the Gaussian space and incorporating current market indicators, order price and volume information can be sampled through denoising. The learned distribution is transparent as it can be directly compared with the empirical distribution and it can be tested for the effectiveness of various conditioning.

Here, the distributions of order price and volume sampled from the DDPM module are examined, and are compared with the respective empirical distributions to evaluate quality. Results are presented in Fig. 6.4 to Fig. 6.8, which show a series of distribution plots for both simulated data (left side) and empirical data (right side) from one trading day of INTC. Identical scale on the vertical axis is set for both simulated and empirical data to allow better comparison. The value range on the horizontal axis for relative price distributions is $\{-2,\ldots,4\}$, and for volume distributions is $\{1,\ldots,79\}$ (values in units of 10^2).

- **The aggregate distribution of price and volume**. As shown in Fig. 6.4, it can be seen that the overall distribution of sampled order attributes highly mimics the empirical data. For the order relative price, it is observed that the probability of $op = 0$ is the highest, and the probability drops quickly for $op > 0$; while the probability for orders that fall within the bid-ask spread (i.e., market orders or aggressive limit orders that form a new best price) is the lowest. For order volume distributions, it is observed that as volume increases, the probability in general shows abrupt downward trend. Both distributions are in accordance

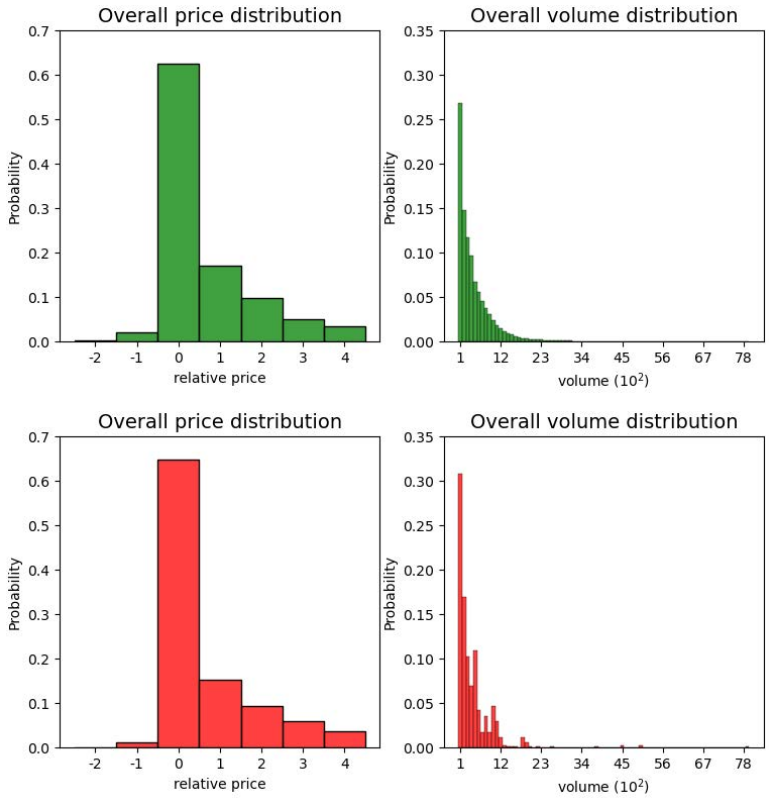


Figure 6.4: The overall distribution for price and volume: simulated (green) and real (red).

with empirical microstructure studies, indicating that the distribution of non-negative order price and order volume follow power law distributions [25]. One point to note, is that even though that the DDPM is used as a non-parametric way to learn the data distribution, the learned probability mass functions appear to be a smooth power law curve (especially for volume distribution), which indirectly reflects the rationality of using a power law to model the distribution of order price and volume in parametric models.

- **Price distribution conditional on event type**. As shown in Fig. 6.5, the conditional price distributions for order submission and order cancellation vary, as cancellation cannot be of a negative price. Additionally, the distribution of non-negative order prices is more skewed for order submissions compared with order cancellations. The distributions for the ask side and the bid side (not shown) are essentially symmetrical.

- **Price distribution conditional on spread**. The bid-ask spread is an important indicator that reveals contemporaneous demand and supply relations. A wider spread indicates a higher degree of investor discrepancy concerning fair asset value. As shown in Fig. 6.6, the price distribution varies significantly when spread enlarges to two, such that almost all orders arrive at the top price levels.

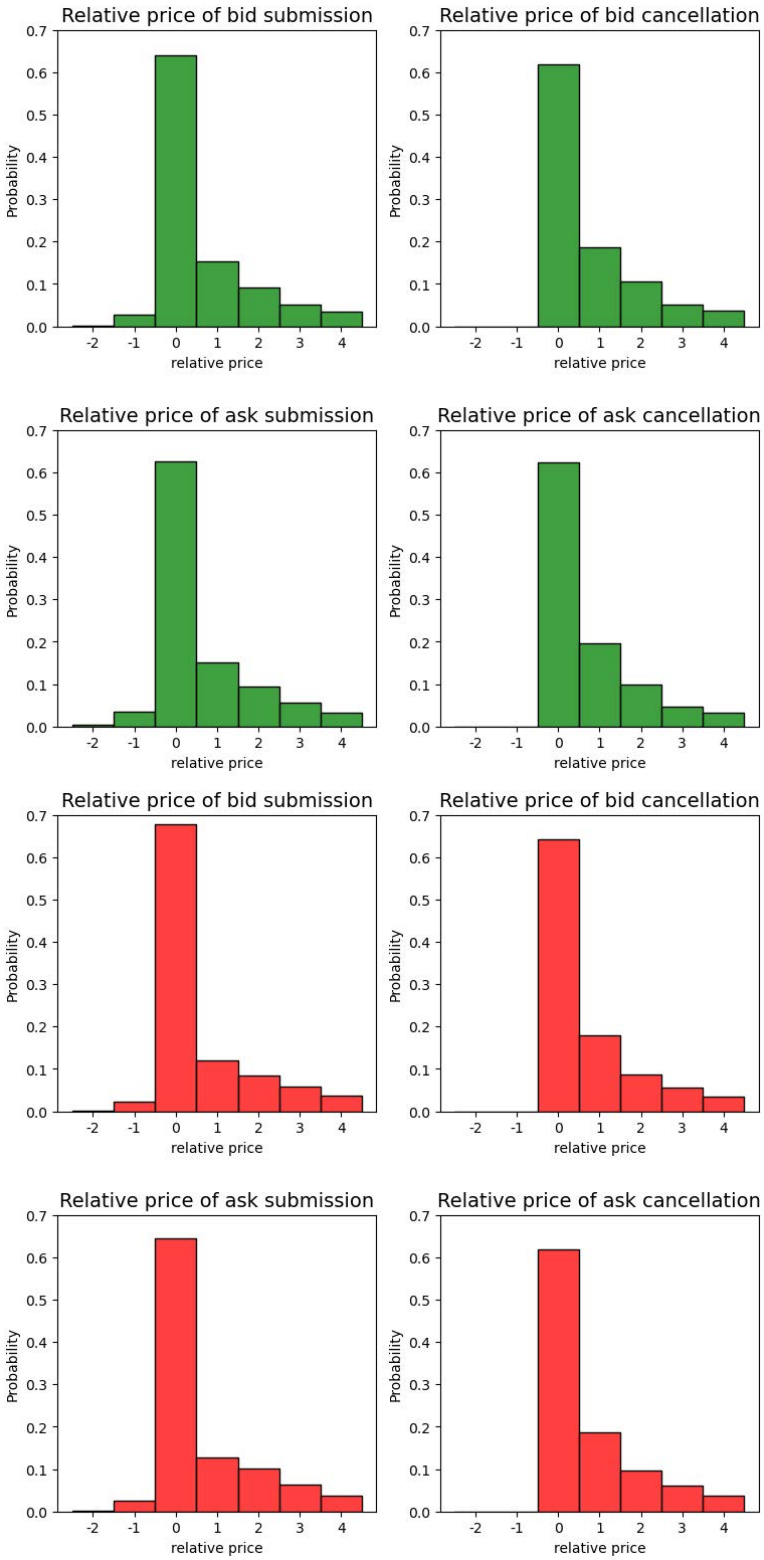


Figure 6.5: Price distribution conditional on event type: simulated (green) and real (red).

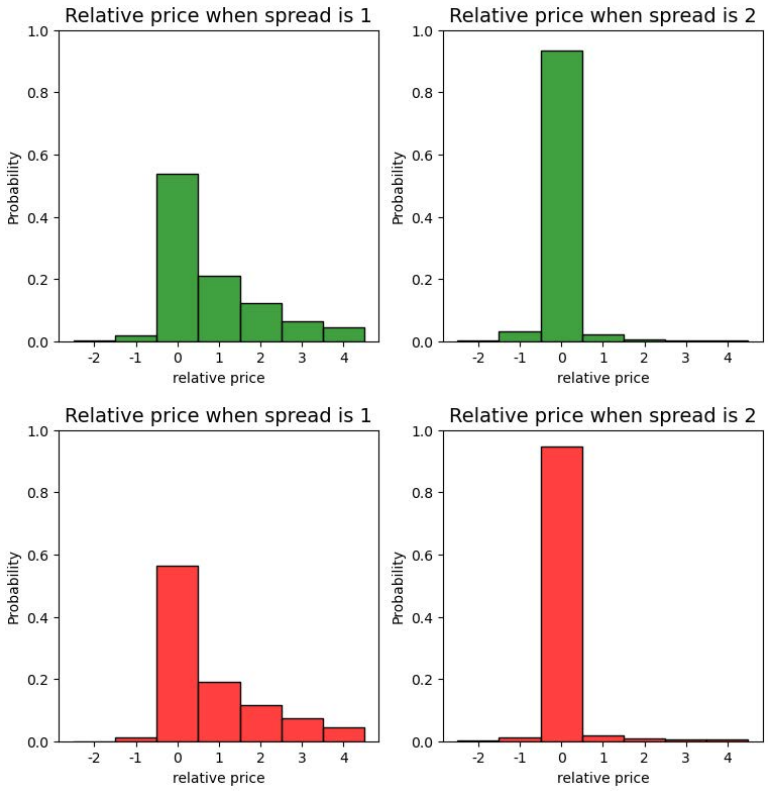


Figure 6.6: Price distribution conditional on spread: simulated (green) and real (red).

- **Order volume distribution conditional on bid-ask volume imbalance**. The difference between the ask side volume and the bid side volume reveals the aggregate market-level imbalance between supply and demand, which ultimately drives the price of the asset to either direction. In a market where demand (total volume of bid orders) exceeds supply (total volume of ask orders), the price is expected to rise and this rise will likely encourage further demand; while the opposite effect exists in a market where supply exceeds demand. As shown in Fig. 6.7, this phenomenon is witnessed in both simulated and real volume distributions: bid orders tend to arrive with a higher volume when demand exceeds supply, and ask orders tend to arrive with a higher volume when supply exceeds demand. This observation indicates that the simulation exhibits market momentum.

- **Order volume distribution conditional on order price**. As shown in Fig. 6.8, the mean volume for limit orders with price -1 (orders that fall within the spread while does not incur trades) is significantly larger than that of orders with non-negative prices. Moreover, for orders with non-negative prices, the right skewness (judged from the coefficient of power law fit on data) of order volume distribution is larger than orders with relative price -1.

- **Dependencies not learned by the model**. In real data, it was found that the arrival frequency of market orders is dependent on the market imbalance. For instance, market ask

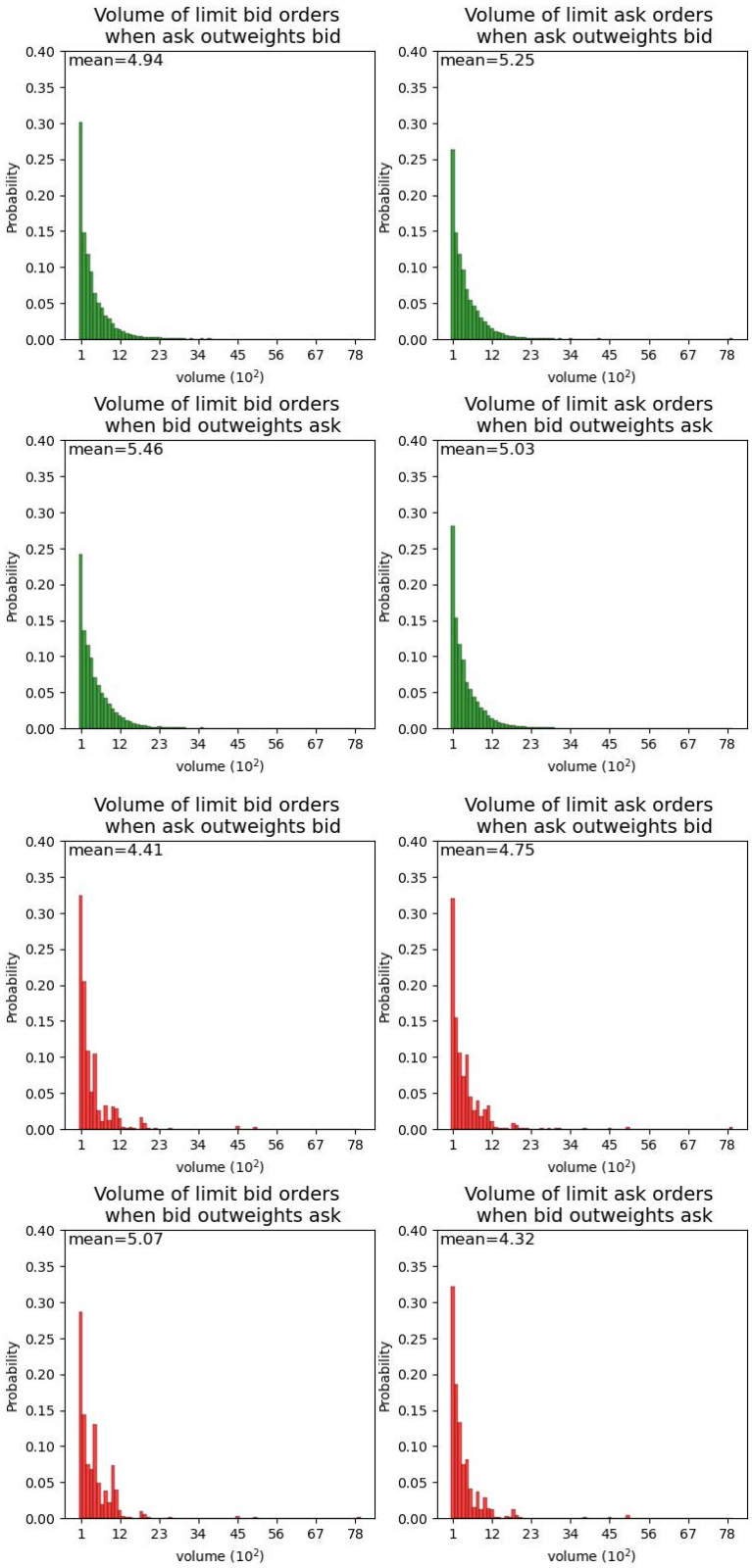


Figure 6.7: Volume distribution for submitted orders conditional on volume imbalance: simulated (green) and real (red).

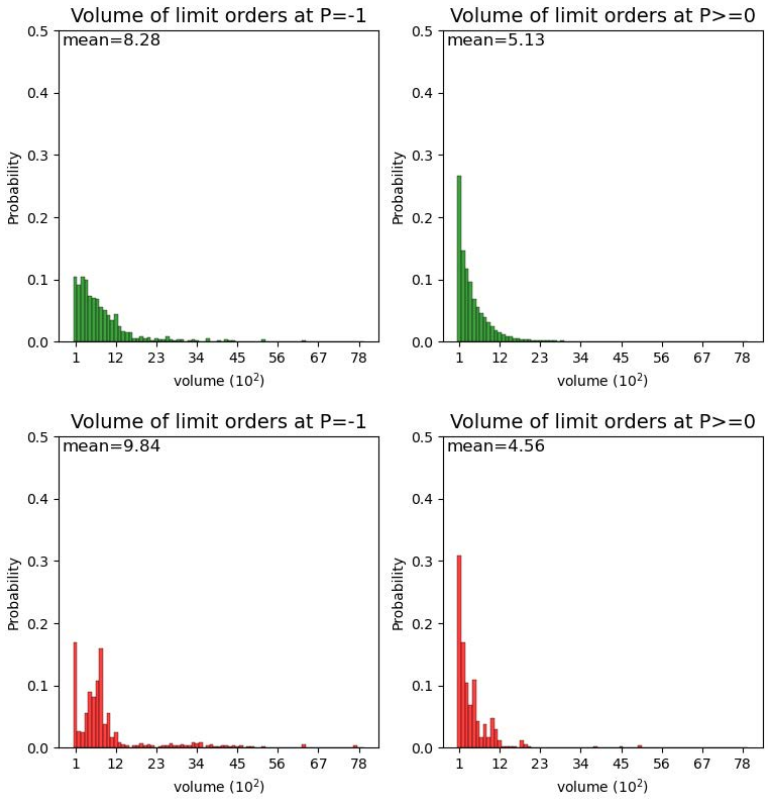


Figure 6.8: Volume distribution for submitted orders conditional on a specific price level: simulated (green) and real (red).

orders arrive more frequently when supply exceeds demand, and vice versa. In comparison, in the simulated data the arrival frequency of market orders does not change when market imbalance varies.

It is worthwhile to note that some of the observations on the empirical distributions might be true only for *large-tick* stocks like INTC, rather than being universally true for all stocks in the market. The advantage of using a non-parametric generative model like DDPM stands out here as it holds no assumptions on the data and can flexibly learn distributions of any form.

6.3.3 Stylised Facts Verification

Stylised facts in economics are empirical findings that are so consistent (for example, across a wide range of instruments, markets, and time periods) that they are accepted as truth. Such facts can be used to verify the fidelity of an economic simulation. In this section, the behaviour of the simulated LOB against a comprehensive list of more than ten *stylised facts* is considered. All order streams are generated by the BT in isolation. Results are gained by evaluating across 20 one-hour samples for both simulated and real data.

The following *stylised facts* were able to be reproduced:

Hurst Exponent for Absolute Return This fact indicates that whether long-range memory exists in financial market time series [42]. A detrended fluctuation analysis [144] can be applied on the time series of absolute returns to calculate the Hurst exponent. A Hurst exponent in the range of $(0.5, 1)$ indicates the existence of long memory. Empirical studies indicate the Hurst exponent to be larger than 0.5 in stock markets [36]. It is found the exponent in simulated data to be 0.60, and 0.56 in the real data.

Autocorrelation in Order-Sign Series This fact indicates that positive autocorrelation exists in order-sign series of submissions and cancellations, respectively. Empirical studies indicate the autocorrelation coefficient roughly falls in the range of $(0.2, 0.3)$ [35]. It is found that the autocorrelation coefficients to be significant in both simulated and real data. In simulated data the coefficient is 0.17 for submissions and 0.16 for cancellations. For real data, the coefficients are 0.41 and 0.33 respectively.

Order Flow Imbalance Impact This fact indicates that the order flow imbalance (OFI) tends to cause prices to change [57]. The imbalance between supply and demand is measured as the difference between events that enforce the bid side (bid submission and ask cancellation), and events that enforce the ask side (ask submission and bid cancellation) during a ten-seconds interval. The R-squared value from the regression between return and OFI was found to have an average value of 0.65 in Cont et al. [57]. Here, the R-squared value is found to be 0.66 in simulated data and 0.70 in real data.

Price Impact Function This fact indicates that the transaction volume's influence on price change is concave. Empirical studies indicate that the impact of transaction volume on change in quote prices increases more quickly with changes at small volumes and less quickly at larger volumes. The slope of the fitted curve between logarithm volume and logarithm price change lies around 0.5 for high market capitalisation stocks, and the slope varies across different markets owing to market protocols [50]. The slope calculated on simulated data is 0.83, and on real data is 0.50, both of which conform with a concave curve.

Mid-Price Evolution This fact indicates that the evolution of the mid-price for liquid financial assets essentially follows a random walk and the mid-price remains volatile during any trading period [12]. As shown in Fig. 6.9a, the mid-price of both simulated and real data fluctuate over an approximately 10 minutes time interval.

Autocorrelation in Return Series This fact, a reflection of the 'efficient markets' hypothesis [145], indicates that price movements for liquid assets do not exhibit significant and strong autocorrelation. The absolute value for autocorrelation of log return time series $f(\tau) = \text{corr}(r_{t+\tau,\Delta t}, r_{t,\Delta t})$ was found to be lower than 0.1, and as time lag increases the coefficient converges to zero [42]. Other studies found that the autocorrelation coefficient for return series to be weak but significant [44]. For simulated data, it is found that the

lag 1 autocorrelation to be around -0.1 and it fast decays as lag increases for one second frequency data. For real data, some samples show weak but significant autocorrelation, while others do not exhibit significant autocorrelation.

Normality of Log Returns This fact indicates that the distribution of asset log returns follows a normal distribution; at the same time, when sampling frequency changes from low to high, the kurtosis increases [146]. Fig. 6.9b shows the distribution of $r_{t,\Delta t}$ when the sampling frequency is high (Δt = 1 sec) and low (Δt = 1 min) can both be fitted with a Gaussian distribution. The distribution of high frequency return shows higher kurtosis and low tails (kurtosis > 10), while the low frequency return shows lower kurtosis and high tails (kurtosis < 1).

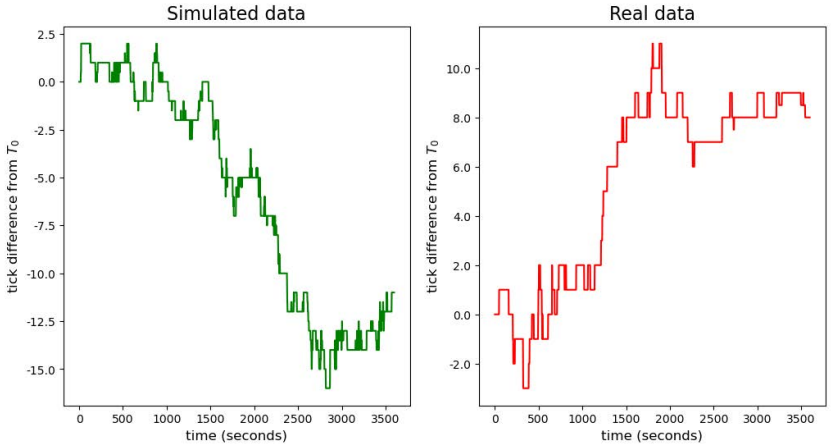
Event Inter-Arrival Time This fact indicates that the distribution of event inter-arrival times follows an exponential or a Weibull distribution [147]. Fig. 6.9c presents the empirical density curve of time distribution (blue line) fitted with exponential (red), Weibull (green), and exponentiated Weibull (orange) distributions using MLE. Both simulated and real data have the best goodness of fit with the exponentiated Weibull distribution (the Jenson-Shannon divergence being 0.22 and 0.27 respectively). Note that the distribution for simulated data rises at its tail, as the maximum value set for inter-arrival time sampling is one second.

Volatility Clustering This fact indicates that high volatility events tend to cluster in time. The function that used to quantify this feature is $f(\tau) = \text{corr}(r^2_{t+\tau,\Delta t}, r^2_{t,\Delta t})$ [42]. This function remains positive and shows a downward tendency when delay in time τ increases. Fig. 6.9d confirms this property is exhibited in both simulated and real data (Δt = 1 sec), and is similar to that shown in Cont [42].

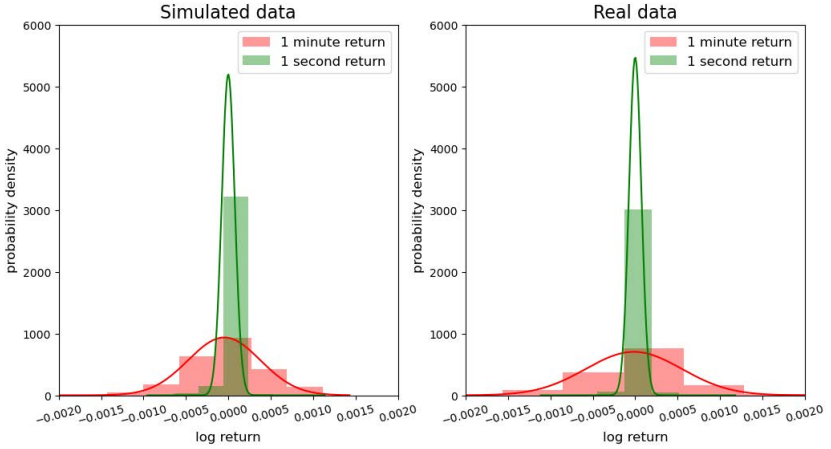
Volatility/Volume Positive Correlation This facts indicates that the standard deviation of log return $\sigma_{\tau,\Delta t}$ and traded volume V_τ have positive correlation, in the sense that trading activities of large volumes are likely to introduce higher volatility [148]. In Vyetrenko et al. [146], the mean value for correlation coefficient for historical data lies around 0.4. This facts is tested for time series resampled every one second, and statistics are calculated for window size of 30 minutes. Fig. 6.9e plots the standard deviation against average trade volume. The correlation coefficient for simulated and real data are 0.35 and 0.62 respectively, which confirm the existence of this fact which confirm the existence of this fact.

The following *stylised fact* is not able to be reproduced:

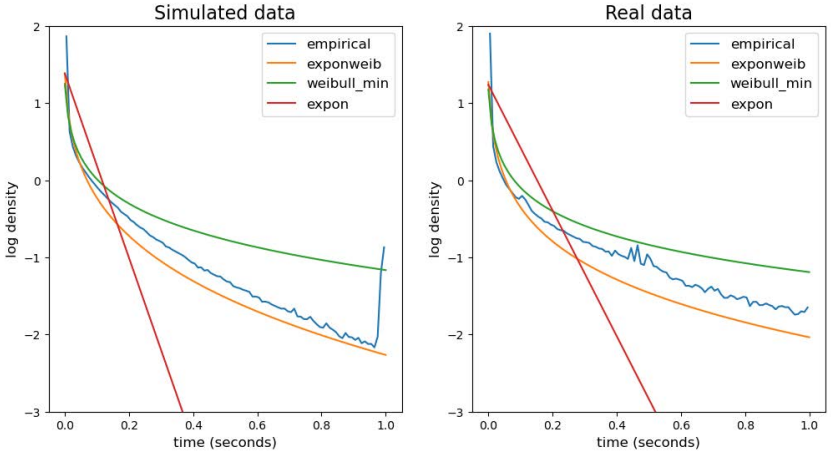
Volatility/Returns Negative Correlation This fact, also termed as the 'leverage effect' [149], indicates that volatility and asset returns are negatively correlated. The volatility commonly



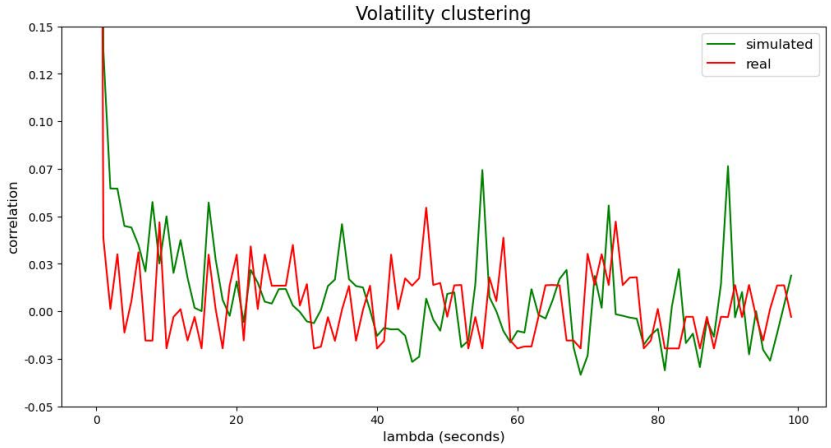
(a) Mid-price evolution: both sim (left) and real (right) mid-price time series exhibit high volatility during an one hour trading period.



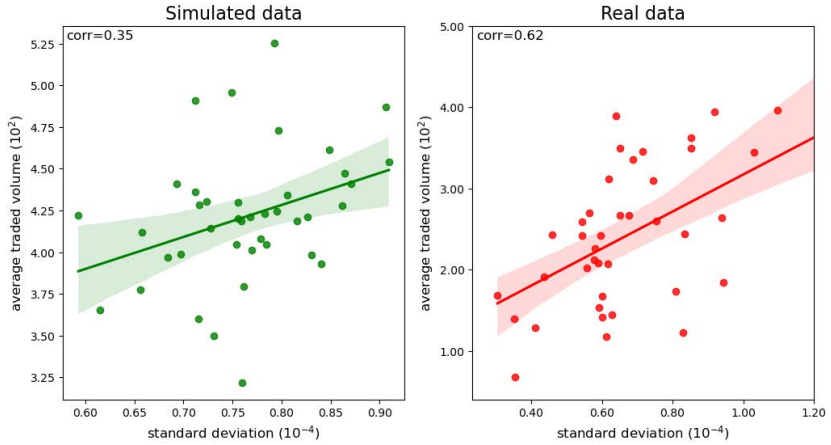
(b) Log returns: Log returns are normally distributed for sim (left) and real (right) data. As sampling frequency increases, kurtosis increases.



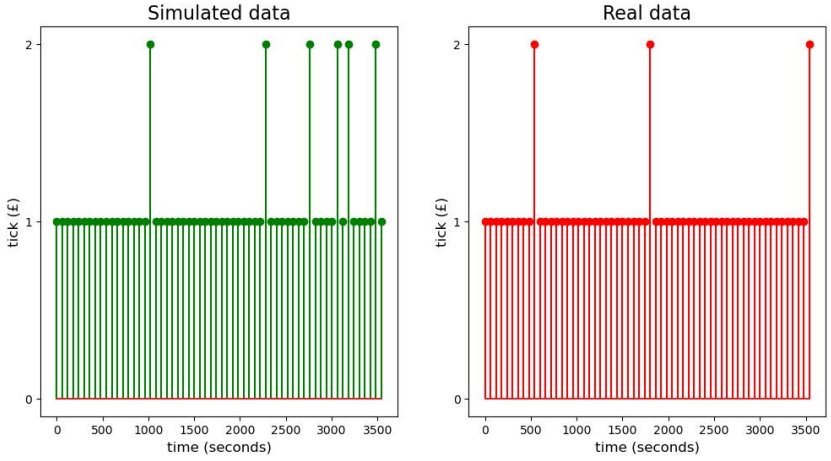
(c) Inter-arrival times: Exponentiated Weibull distribution is found to be the best fit for both sim (left) and real (right) data.



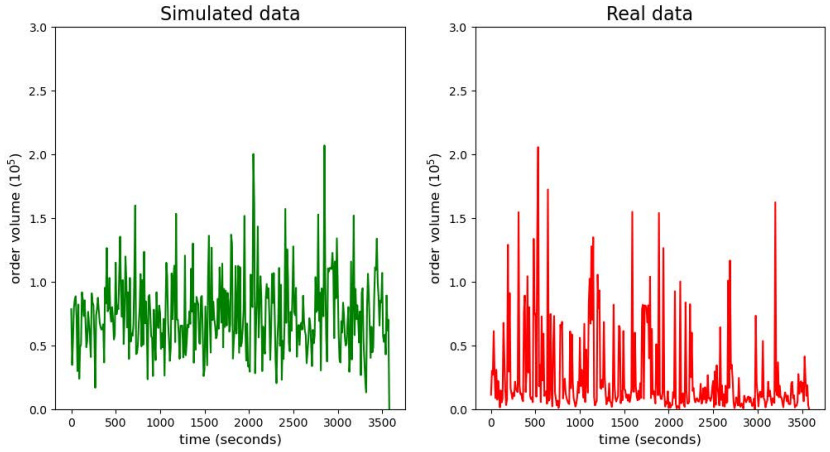
(d) Volatility clustering: sim (green); real (red). Correlation for absolute return over short time lags, decaying to zero as lag times increase.



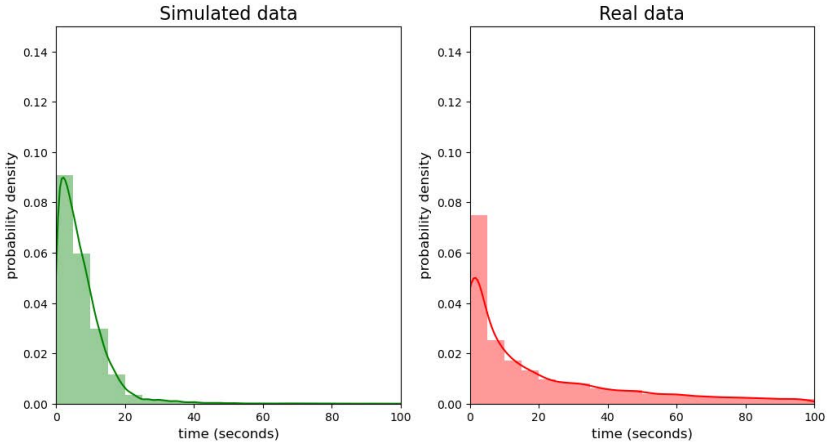
(e) Volatility-volume: Positive correlation of returns standard deviation and average traded volume in sim (left) and real (right) data.



(f) Bid-ask spread: both sim (left) and real (right) data have bid-ask spread of one tick for majority of time, showing high liquidity and large-tick stock essense.



(g) Incoming order volume: both sim (left) and real (right) data show volatile incoming volume, with sim showing higher mean and lower variation.



(h) Time to first fill: both sim (left) and real (right) data show power law distributions, with sim showing a shorter tail.

Figure 6.9: Stylised facts and numerical properties for both simulated and real LOB data.

used here is the implied volatility derived from a particular volatility index (e.g., VIX) or the price of financial derivatives. Here, there does not exist an implied volatility for the simulated asset. Therefore, following Dufour et al. [150], The relation between realised volatility (denoted as the sum of squared log returns) and returns is investigated. It is found that both simulated and real data do not show a significant negative correlation between the two variables. This may suggest that realised volatility is not a good substitute of implied volatility. Although the leverage effect has previously been demonstrated using an ABM [151], it is rare. In Chen et al. [151], it was suggested that the reason many ABMs are not capable of recreating this fact is potentially due to the lack of asymmetric trading behaviours in simulation.

By taking reference from Coletta et al. [111], some numerical properties of the simulated LOB are also exploited. Such properties vary a lot across different financial markets and financial assets, and they normally cannot be concluded with certain distributions with an agreed-upon parameter range. Hence, The simulated data generated by the BT agent is compared with the real data based on which the BT agent is trained, instead of conducting a direct comparison with Coletta et al. [111].

Bid-Ask Spread over Time Shown in Fig. 6.9f, as both the simulated data and the real data is for highly liquid stocks that have comparatively low market value, the bid-ask spread is dominantly one tick over time. Only when all orders resting on the top price levels are taken by market orders, or cancelled by agents, the bid-ask spread can be larger than one.

Incoming Volume for Limit Order Submissions Liquid financial assets usually have high limit order incoming volume, and the volume fluctuates over time. From Fig. 6.9g it can be seen both simulated and real data present high incoming volume and high volatility, with the simulated data exhibiting comparatively higher mean value and lower relative variation. The differences mainly originate from the setting of a stable distribution form for limit order submissions in the simulation, while in real markets the distributions of such properties can be dynamic over time according to market conditions and cannot be concluded with a stable form of distribution.

Time to First Fill Time to first fill describes the time interval between the time when a order is submitted and the time when the order is partially fulfilled. This property indicates whether an asset is liquid or not, and liquid assets usually have low time to first fill. Here the majority of first time to fill for both simulated and real data are less than one second, as in Fig. 6.9h. Empirical research also indicated that the time to first fill follows a power law distribution [147], here both simulated and real data are in accordance with the literature (KS test p-value <0.1)

6.3.4 Experimental Agents Interaction

6.3.4.1 Interaction Experiment Settings

By incorporating agents with various trading strategies into the ABIDES framework to interact with the BT, insights into how the system reacts to external stimuli, and how those reactions compare to real markets can be gained. Here, five criteria are considered: (i) mean profitability of agents (with no consideration of transaction cost); (ii) trading volume of agents as a proportion of total market trading volume; (iii) standard deviation of the realised mid-price time series; (iv) BT's ask-bid order imbalance: measured as $\max(AS+BC, BS+AC)/\min(AS+BC, BS+AC)-1$, in which AS, AC, BS, BC indicate the number of ask submission, ask cancellation, bid submission

Table 6.1: Agents and LOB statistics in markets containing one BT and n homogeneous trading agents of type T. Where n is shown in parentheses, BT is set to have no order flow impact. Criteria marked * are in %.

	(i) Profit (in 10^{-3})				(ii) Prop. trades*				(iii) Std. mid-price				(iv) Order imbalance*				(v) Correlation			
T \ n	1	15	(15)	50	1	15	(15)	50	1	15	(15)	50	1	15	(15)	50	1	15	(15)	50
MM	1.6	12.1	0.2	55.5	1.2	11.7	15.5	24.0	4.3	9.9	5.8	28.0	1.1	3.5	1.1	9.7	0	0	0	0
MR	-1.0	0.8	-1.0	1.2	1.3	16.1	17.2	36.0	3.7	2.7	3.5	2.9	1.0	1.1	1.2	0.9	0	0	0	0
ZI	-0.1	0.5	-0.5	1.0	0.6	6.0	6.7	14.4	4.9	3.8	4.1	3.3	1.3	0.9	1.2	1.0	0.19	0.27	0.05	0.76
HBL	-1.1	0.8	-0.1	0.9	0.6	7.3	7.5	18.6	4.6	3.6	3.5	3.1	1.1	1.0	1.1	0.9	0.09	0.44	0	0.82

and bid cancellation; and (v) the correlation between *value* agents' underlying fundamental stock value and realised stock price.

Each experimental condition is repeated 20 times and mean values are taken. Results are shown in Table 6.1. Number of agents is set as $n \in \{1, 15, 50\}$, and order flow impact *True* (i.e., BT reacts to agent orders). For control, markets with $n = (15)$ agents are also compared with order flow impact *False* (i.e., BT does not react to agent orders). Unless otherwise stated, agent populations are homogeneous.

6.3.4.2 ABIDES Configuration

Four types of agents are considered: MM, MR, ZI, and HBL. To enable fair comparison, all agents wake up according to an exponential scheme (Poisson process), with average interval $\Delta t = 30secs$. The order events posed by experimental agents as a percentage of all order events is below 10% in all markets, ensuring that the market dynamics created by the BT is not overtaken by experimental agents. Other parameters are: $l_1 = 20$, $l_2 = 50$, $r_{max} = 5$, $l = 16$. For ABIDES related configurations, the Reference Market Simulation Configuration sample file is considered. Values are set as: mean value $\mu = 10^3$, $\sigma^2 = 2 \times 10^{-10}$, reverting rate $\gamma = 10^{-12}$. The original default values are: 10^5, 10^{-8}, 1.7×10^{-16}, respectively. The values of μ and σ^2 are lowered because of a lower starting asset value £10. The value of γ is increased so that the oracle demonstrates more mean reversion during 1-hour simulation window. Observation variance is set as $\sigma_o = 10$. $R\_min = 0, R\_max = 5, eta = 1$ are used to control the greediness of agents. Agents' holding limit and starting cash are set big enough as that their decisions are not bounded by them. All findings to follow are supported by Wilcoxon signed-rank test (p-value<0.1, unless otherwise specified) either by comparing statistics in group $n = 1$ and group $n = 50$, or by comparing group $n = 15$ and group $n = (15)$.

6.3.4.3 Herding Effect

Financial herding has been well documented in market empirical studies as an phenomenon of investors tending to follow the crowd or trend in the market instead of performing their own

analysis [152, 153]. Here, making reference to Table 6.1, the experiment considers how a change in the number of agents holding the same trading strategy affects agent profitability and LOB behaviour.

In terms of agent profitability (i), it can be seen that in general the mean profitability increases with number of agents in markets containing both *trend* and *value* strategies. Empirical studies on emerging and less-efficient markets indicate that the intensity of herding is positively related to trading profitability, especially for *trend* strategies [154, 155]. Studies that investigated the profitability of several technical trading strategies also revealed that by increasing the frequency of the data that the strategy is based on, the profitability of the trading strategy can be improved [156]. Also, as more agents adopt a particular *trend* strategy, such as momentum trading, there is greater influence on market price in a favourable (i.e., predictable) direction. Among all agents, MM achieves the highest profits. This profit is mainly derived from the capital gains in stock positions after causing an extreme one-direction price movement, as MM agents rarely neutralise a position after momentum ignition.

In terms of LOB volatility (iii), it can be seen that increasing the number of MM agents causes increasing volatility, while the opposite effect occurs for other agent types. This is to be expected for momentum traders as price movements drive further movements in the same direction. Empirical studies of real markets indicate that herding behaviour is more intense in extreme market conditions, caused by momentum trading behaviours, and can lead to tail events such as a market crash [157]. Boehmer et al. [157] argued that homogeneous trading behaviours are more likely to cause price overreaction during a short time period, with deviation from the long term mean. In contrast, MR agents and *value* agents follow a mean-reverting fundamental value oracle. Therefore, an increase in the number of agents tends to reduce market volatility.

In terms of correlation between fundamental price and realised LOB price (v), empirical studies have indicated that the high intensity of HFTs' herding behaviour tends to increase the correlation of their order flows [158], giving them more power to move the price. Studies on the futures market also indicated that HFTs help prices converge to the fundamental [159]. As *value* agents trade according to the comparison between their underlying fundamental value and the realised stock price, increasing the number of agents essentially gives the group more power to move price towards the fundamental value. The effect of this herding behaviour is reflected in the increased correlation for ZI and HBL. Notice that markets containing only *trend* agents exhibit zero correlation between the fundamental value and realised LOB price. This is to be expected as the fundamental value is ignored by these traders.

6.3.4.4 Order Flow Impact

The statistic of order imbalance generated by the BT agent (iv) can be used to determine whether agents impact order flow imbalance. By comparing $n = 15$ with the control $n = (15)$, it can be clearly seen that the degree of imbalance is significant in MM market, while not significant in

markets with other types of agents.

In markets dominated by *trend* agents, the order flow impact manifests as causing further trend following events (i.e., pushing price farther away, or pulling price back towards the mean). This effect is observed in the volatility of mid-price (iii). When no order flow impact is considered, volatility in a market full of MM traders tends to be underestimated. The underestimation of volatility in a MM market is caused by overlooking the empirical finding of momentum ignition [160]. Momentum ignition indicates that investors tend to follow the price trend made by HFTs.

Studies of real markets indicate that large institutional *value* traders contribute substantially to price discovery [161]. A similar effect is observed in the simulation results. It can be seen that an increase in the number of *value* agents in the market causes an increase in the correlation between realised LOB price and the fundamental value (v). Also, correlation is nearly zero in the control, where order flow impact is excluded. This result indicates that the inclusion of order flow impact can help model the price discovery role of *value* agents.

6.3.4.5 Competition between Agent Strategies

To understand strategy interaction, heterogeneous experiments are also performed, with the existence of order flow impact: (1) *trend* markets containing 15 MM vs. 15 MR; and (2) *value* markets containing 15 ZI vs. 15 HBL. In *trend* markets, it is found that the profits of both strategies are inferior to the profits generated in homogeneous markets, as these two types of agents are competing with each other to influence the price in their respective favor. Also, the resulting market volatility falls by nearly 50% compared with a homogeneous MM market, indicating MR agents' mean-reverting impact. In *value* markets, profits of both strategies are not significantly different with their respective homogeneous markets. However, in relative terms, HBL is more profitable than ZI. This is unsurprising given HBL has a relatively sophisticated trading strategy, when compared with ZI.

6.3.4.6 Stylised Facts with Interaction

Finally, the *stylised facts* of simulated markets where BT interacts with agents are again measured. Homogeneous markets containing each trader type, with $n = 15$ and order flow impact set *True*, are considered. Results are: (1) Hurst exponent: $[0.58, 0.76]$; (2) Autocorrelation of order signs: $[0.16, 0.19]$ for submission and $[0.15, 0.18]$ for cancellation; (3) OFI R-squared: $[0.62, 0.68]$; and (4) Price impact function slope: $[0.74, 0.88]$. In addition, other non-qualitative properties are also in accordance with the findings in the BT standalone experiments.

6.3.5 Responsiveness of the System

Here, the approach in Coletta et al. [111] is closely followed to test the responsiveness of the system to price impact. A percent-of-volume (POV) agent is introduced which simulates a large

volume trader that is likely to move the market price. The POV agent is defined by two parameters: time interval T seconds, and percentage of market volume λ. At time t, POV is instructed to buy/sell a total volume $V = \lambda M$, where M is the total transaction volume of the whole market over the previous T seconds. POV will attempt to trade volume V over the next T seconds by splitting V into multiple smaller orders.

6.3.5.1 Price Impact by POV Agents with Various Trading Volume

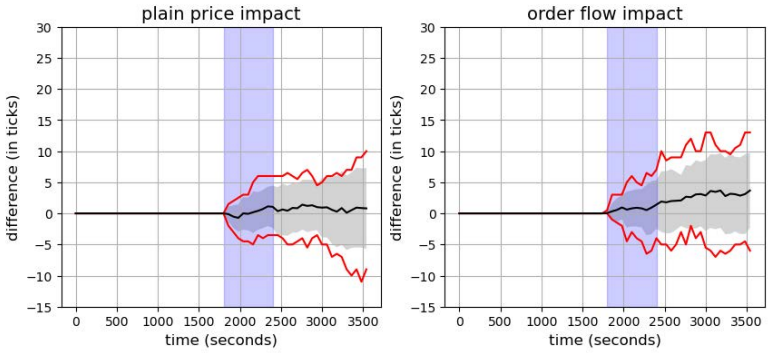
Experiments are set to run for one simulated hour and POV agents are set to begin trading at time $t = 1800$ seconds, with trading interval $T = 600$ seconds; i.e., POV will do nothing for the initial 30 minutes, followed by a burst of trading between 30-40 minutes, and then no further trading until the simulation ends. The price impact of increasing trade volumes is explored, such that $\lambda \in [0.01, 0.1, 0.2, 0.5]$.

For each set of experiments, the simulation is set to run under three conditions: (1) A market containing BT agent only. Here there is no POV agent and so the evolution of the market when there is no impact can be traced; (2) A market containing POV agent and BT agent configured to have no *order flow* impact such that the BT does not react to the exogenous orders posted by the POV. Here, any price change in the market is a direct result of POV order submissions only; (3) A market containing POV agent and BT agent configured to have *order flow* impact. In this system, changes in market price result from both the POV order submissions and also the responsive market behaviours of the BT.

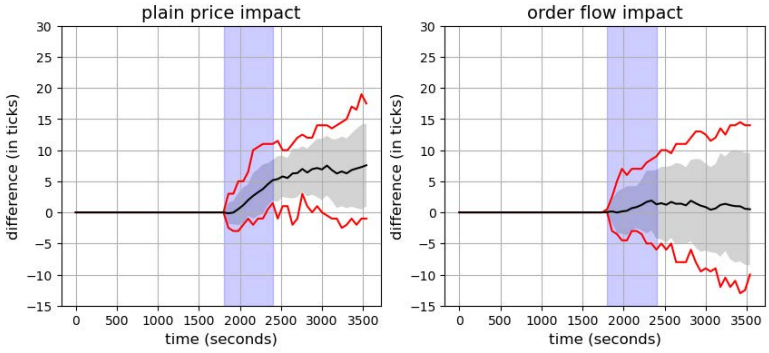
By comparing the difference in market price between configurations (1) and (2), it is possible to investigate the price impact of simply adding orders into the market and this price difference is referred to as *plain* price impact. This impact resembles traditional backtesting, where orders eat the book and are not replaced as the market cannot respond. By comparing the difference in market price between configurations (2) and (3), it is possible investigate how the market responds to POV order volume. This 'extra' price impact is referred to as *order flow* impact. This approach of separating total price impact into two components is unique, and different to the approach taken by Coletta et al. [111].

Results are shown in Fig. 6.10, with the left hand side showing *plain* impact calculated as the difference in mid-price generated in (1) and (2), and the right hand side showing *order flow* impact calculated as the difference in mid-price generated in (2) and (3). The black line, red line, and grey shaded area are mean value, 10-th and 90-th percentiles, and one standard deviation from the mean, respectively. The POV agent submits orders during the blue shaded area. Each simulation is repeated 20 times, and the POV agent only submits bid orders.

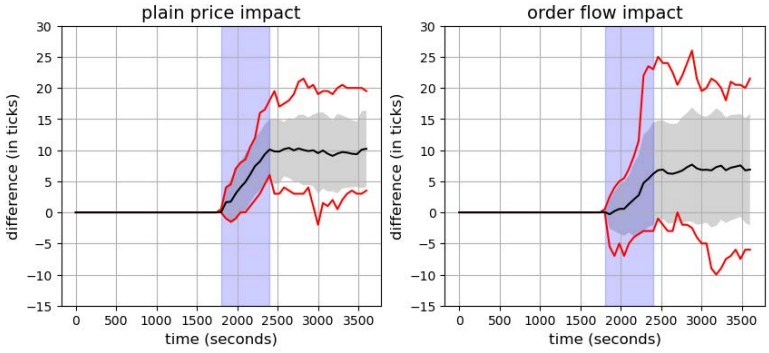
For *plain* price impact, it can be seen that price rises when POV trades, and grows monotonically with λ. Price impact is permanent and remains after the POV agent stops trading. For *order flow* impact, it is found that low values of $\lambda = 0.01$ and $\lambda = 0.1$ do not cause the market to respond. As λ increases, *order flow* impact increases superlinearly. This shows the market adversely



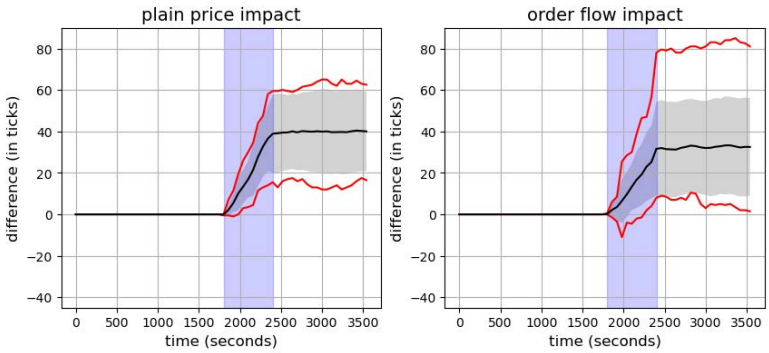
(a) Mid price difference when $\lambda = 0.01$



(b) Mid price difference when $\lambda = 0.1$

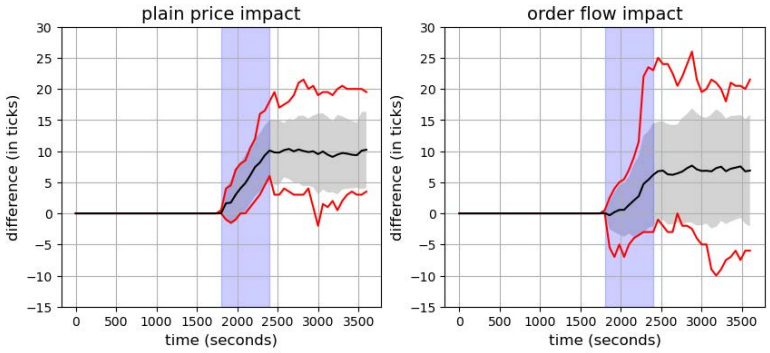


(c) Mid price difference when $\lambda = 0.2$

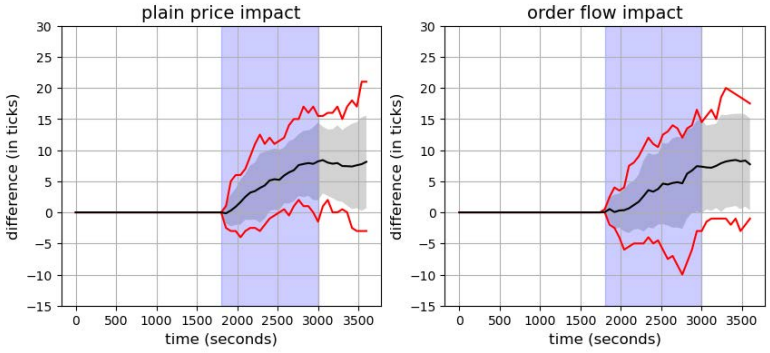


(d) Mid price difference when $\lambda = 0.5$

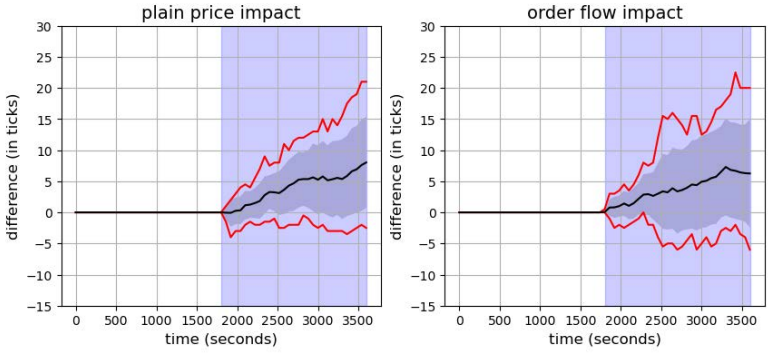
Figure 6.10: *Plain* price impact and *order flow* impact as percentage of volume λ varies.



(a) Mid price difference when trading horizon is T



(b) Mid price difference when trading horizon is 2T



(c) Mid price difference when trading horizon is 3T

Figure 6.11: *Plain* price impact and *order flow* impact as trading horizon varies when $\lambda = 0.2$.

responding to increased buy pressure, pushing market prices higher than they would otherwise go. Noticeably, the *order flow* impact is the overlooked portion of price impact in traditional backtesting, which assumes the market does not respond to exogenous events.

The empirical study by Bershova and Rakhlin [162] showed that after the instant price impact caused by trade T, the price presents a mean reversion pattern of moving back towards the original price before trade T executed. Such a phenomenon is called 'post-trade price reversion', and the reversion was found to be 1/3 of the instant price impact. In the simulation results shown in Fig. 6.10, price impact does not show post-trade reversion. For comparison, in the related

simulation experiments performed by Coletta et al. [111], the price time series either presented full reversion for small λ or no reversion for large λ. Therefore, both methods are unable to accurately capture this phenomenon. The reason for NS-ABM's failure in replicating this price dynamic might be that both the sd-PNHP and the DDPM module do not have knowledge of past price information, and as a result the price reversion towards a historical level cannot be properly modeled. This limitation will be addressed in future work.

6.3.5.2 Price Impact by POV Agents with Various Trading Horizons

This section further investigates how price impact responds to rate of trading (i.e., holding volume constant and varying trading horizon). As indicated by optimal execution theories, a higher execution rate (or shorter trading period) tends to cause greater market impact [163]. To test this observation, a final experiment is performed using POV trader with $\lambda = 0.2$ and total volume is set as $V = \lambda M$, where M is the total transaction volume of the whole market over the previous $T = 10 \times 60$ seconds. Three configurations are compared, such that the POV must trade total volume V within T, $2T$, and $3T$ seconds.

Results are shown in Fig. 6.11. By adding up both the mean values of *plain* price impact and *order flow* impact at the end of POV trading periods (i.e., where blue shaded areas end), the price differences are 16.3, 15.6 and 14.3 for trading horizons of T, $2T$ and $3T$ respectively, demonstrating a downward trend. This conforms with the observations of Almgren [163]. However, other empirical studies have reached the contradictory conclusion that market impact tends to increase for a longer trading horizon [164]. Thus, this topic remains to be further investigated.

6.4 Conclusions

This chapter presented the hybrid neural stochastic agent-based model (NS-ABM) for realistic LOB simulation implemented in the ABIDES framework [138]. NS-ABM combined the benefits of ABM with data-driven approaches to simulation by including a neural stochastic BT that is pre-trained on real data. The backbone of the BT consists of a sd-PNHP module for event type and arrival time sampling, and a DDPM module for order-related attributes generation, results from which are of high interpretability and transparency. The BT trader has been shown to realistically simulate real world LOB dynamics, with ten *stylised facts* – empirically observed properties of markets that are accepted as fact – approximately reproduced. In addition, the BT trader can also realistically react to endogenous market events, with *stylised facts* remaining once a populations of *trend* and *value* trading agents are added to the simulation. An explicit investigation of the price impact within the system was also conducted, and it was elaborated how it mimics and deviates from empirical studies. Since the NS-ABM model can realistically replicate market characteristics, it removes the need to include populations of stochastic 'noise' agents that approximately characterise market dynamics (e.g., McGroarty et al. [10]). NS-ABM

can also act as a 'dynamic back-test' harness, such that individual trading strategies can be evaluated on historical data that realistically adapts to the actions of the trading strategy.

In the experiments, an exogenous fundamental value was generated using a mean-reverting stochastic process. However, ABIDES enables historical data to be used as a fundamental value. Using such a configuration, the NS-ABM offers a potential route towards the 'holy grail' of dynamic back-testing for financial trading algorithms, where trading events generated by a strategy under test have trading impact. It would also be interesting to conduct more diversified interaction experiments, for instance simulations that include more heterogeneous trading strategies, to see how strategies interfere with each other and how the resultant market statistics revolve. In terms of the BT, it is also interesting to investigate the feasibility of replicating structural changes (e.g. a market crash) that embedded in the training data, which provides the possibility of replicating more realistic, or even abnormal, market dynamics in LOB simulation. This exciting avenue of investigation is intended to be explored in future work.

CHAPTER 7

CONCLUSION

This thesis has presented a comprehensive series of theoretical studies on modelling the LOB, utilising state-of-the-art deep learning models. The motivations of the studies conducted were mainly coming from realistic dilemmas in research that involves LOB. First, LOB data results from the interaction of orders that are submitted to the exchange, or on a high level the strategic competition between market participants. In this sense, the information embedded in the LOB is valuable, as it reflects the contemporaneous demand and supply relation in the market, which in the end becomes the essential driver of market price for an asset. This characteristic of LOB data decides its cornerstone role for researchers that are focused on financial microstructure studies, as well as market participants that want to improve profitability of their trading strategies. While on the contrary, the availability of LOB is very limited or comes with a high cost, owing to the streaming, storage, cleaning and reconstructing fees associated with processing the extreme large volume of high frequency LOB data. This poses an obstacle for people interested in exploring the data. Second, there is urgent needs for a dynamic interactive LOB simulation system. For researchers, it is essential to test and validate the generality of their empirical findings found on the real market, and conduct experiments like sensitivity analysis to investigate the casual relations in market variables; For market investors, they need to test the profitability and risk profile of their trading strategies to gain insights of how would it perform in the real market. Their are two general ways to get the intended works done, one is testing on the historical LOB data (i.e., backtesting); the other alternative is sorting to market simulation. Unfortunately, both methodologies have their unique and unneglectable drawbacks. For testing on the historical data, the unrealistic assumption of a static market, a market that will not react to the change in some market variables or actions posed to the market, is held. This assumption is dangerous as it disobeys the reality, as the market absorbs, digests, and react to those changes in real time,

suggested by the efficient market hypothesis. On the other hand, sorting to LOB simulations becomes a better alternative, as it can be designed to be a dynamic system that is responsive to external stimuli. Such a LOB simulation in most cases comes in the form of agent-based model (ABM), a simulation environment mimicking the real market mechanism in which interaction between strategic agents, together with some market rules, decides the dynamic of the system. It may seems a perfect solution while in reality it is not the case. The major concern of this methodology is that almost all existing ABMs are governed by subject agents whose behaviour patterns and parameters are defined by human beings. Simulating the LOB under this manner inevitably encounters subjectivity issues, as the existence of those agents in the real market is questionable. Even their existence might be confirmed, the behaviour patterns of the agents may not really be in accordance with the real world. Thus, this thesis has been devoted to solving those dilemmas aroused from real world application scenarios.

7.1 Main Contributions

To allow a broad view, this thesis concentrated on three research topics: (i) From trades and quotes (TAQ) data to LOB: As top-level TAQ data is much more accessible than LOB, and TAQ data itself is an overwhelmingly important component of the LOB, the thesis attempted to recover the volume information on deep price levels of the LOB using TAQ data; (ii) From event stream to LOB: As indicated by empirical studies that the event stream in financial markets presents memory, the thesis attempted to predict information concerning the next coming event based on historical event stream. Facilitated by stochastic point sampling method, a minimal LOB simulation was achieved by iteratively sampling new events and accumulating sampled events to create an event-based LOB; (iii) Hybrid neural stochastic agent based LOB simulation: By embedding the models which can make accurate predictions on the attributes of the coming event as a background trader (BT) in an ABM simulation environment, the thesis was able to construct a realistic LOB simulator that is both grounded on real data and being interactive with experimental agents. In the following, the main contributions of the thesis are discussed in details:

- Chapter 4 presented the LOB recreation model (LOBRM). The LOBRM is able to recover deep level information of the LOB based on purely top level TAQ data. The research first proposed a sparse one-hot positional encoding method for TAQ that can efficiently encode TAQ data into feature vectors which can be better understood by deep learning models. The main working components of the LOBRM is of three modules: (i) a history compiler that makes predictions from a historical perspective, based on the volume resilience characteristic of the LOB; (ii) an events simulator that makes predictions from a dynamic perspective, simulating the incremental portion of the prediction as the results of stochastic processes; (iii) a weighting scheme that adaptively combines the basis prediction and incremental

prediction. The core component of the LOBRM is an ordinary differential equation recurrent neural network (ODE-RNN), a variant of RNN that is capable of modelling irregularly sampled time series by constituting a continuous latent state. Through experiments, it is shown that the LOBRM is superior than mainstream time series models in this task facilitated by continuous-time RNN and effective ensembling, achieving an mean absolute error ranging in 0.3 to 0.6 multiples of volume standard deviation in prediction. Further experiments explored the generality of the encoding method for TAQ and factors that are related to the model performance.

- Chapter 5 presented the state-dependent parallel neural Hawkes process (sd-PNHP) for LOB event stream prediction and simulation. The research was rooted in the traditional empirical and theoretical LOB studies that modelled the event stream in the LOB as stochastic point processes. The sd-PNHP differs from the original neural Hawkes process in two ways. First, in order to further improve the model's performance in handling multivariate point processes, a stacked structure of continuous-time long short-term memory (CT-LSTM) units design that provides fine-grained modelling for intensity rates of different types of events was proposed. Second, considering the complexity in financial markets, an event-state interactive mechanism was implanted in the model. Within the interaction, the state (i.e., market indicator) influence the intensity rates, and in turn the arrived events influence the Markovian transition of the state. Through experiments, the sd-PNHP was shown to prevail other mainstream models, achieving a prediction accuracy ranging in 50% to 60% in the formulated four class classification task. By embracing the unique advantage of neural point process models, a minimal simulation of the LOB was achieved by stochastically sampling events from the well-trained sd-PNHP model. A sensitivity analysis was provided to gain a better insight of the causal relations between stochastic properties of the LOB and input factors in the system, paving the path for further exploration of the sd-PNHP's potential in LOB simulation.

- Chapter 6 presented the hybrid neural stochastic agent-based model (NS-ABM) for LOB simulation. As the final goal of the thesis is to create a realistic LOB simulation environment that is both grounded on real data and being interactive, the research combined the advantages of stochastic and agent-based modelling of the LOB to come up with a hybrid model. First, a BT whose actions mimic the aggregation effect of the whole market was designed. For this agent, the event type generation was governed by the sd-PNHP, and other order related attributes were governed by the conditional deep diffusion probabilistic model (DDPM). Both logic modules were strictly trained and validated on real data, enhancing its objectivity. Then, the BT was introduced to a mainstream ABM simulation platform for LOB. When the BT was running on its own, it could be deemed as a complete market; and when there were experimental trading agents activated to join the simulation, the BT could make real-time responsive actions to the experimental agents' behaviours and

hence deviated from its original simulation trajectory. Through experiments, it was shown that the simulated LOB exhibited a comprehensive list of *stylised facts* found in real data, demonstrating its authenticity. Further interactive studies showed that the interactions made by the BT to experimental strategic traders are in accordance with findings in the real market, and the resulted price impact from interactions was found convincing when compared with empirical studies.

To conclude, in terms of modelling and simulation of the LOB, the thesis revolutionised the traditional methodologies by taking advantages of some of the most advanced deep learning techniques. The thesis investigated the relations between several data types in the LOB, and achieved realistic LOB simulation based on the gained insights. The proposed methodologies provided appropriate solutions to the aforementioned dilemmas in application, by recreating LOB from TAQ, predicting information concerning next LOB event arrivals, and conducting realistic LOB simulation using a hybrid model. The models chosen were well suited for the nature of the questions under investigation, providing a brand-new perspective in LOB modelling. Research on the LOB is abundant while some specific fields are still on-going and fast-developing, thus these new perspectives provided in this thesis can hopefully be inspirational for further research.

7.2 Limitations

The research conducted in this thesis is not without limitations. This section first discuss the limitations shared by all the research conducted as a whole, and then dive deep into the limitations of each model proposed.

There are mainly three high-level limitations for research conducted in the thesis.

First, the dataset used by all three research conducted is comparatively small, being a dataset consisting of five days length LOB for three *large-tick* stocks. Even though the dataset contains roughly five million LOB updates, there is still possibility that the dataset being a biased representation of the real data distribution, limiting the models generality. The main reason for not incorporating a larger dataset is that there is no such structured and complete LOB dataset freely accessible in the public. There indeed exists some large LOB dataset [124], while the data is reconstructed and missing key components that involved in the studies conducted in this thesis. This limited availability of a complete LOB dataset again echoes with the motivation of the thesis - increasing the availability of LOB data using deep learning techniques. To conclude, this limitation is inevitable.

Second, the models proposed in this thesis were all designated for the LOBs of *large-tick* stocks with five price levels, which naturally simplified the modelling process. *large-tick* stocks have orders densely distributed around quote prices, leading to the assumption of one tick size intervals between price levels on both sizes of the book. In reality, the structure of the LOB for some of the stocks can be much more complicated. For instance, the intervals between price

levels in the LOB can be irregular for high-priced stocks. In this scenario, the assumptions of the models do not align with data and the models may fail to work. Still, the thesis provides a good starting point to be further extended.

Finally, the time complexity of the proposed models is considerably higher than traditional machine learning baselines or discrete RNN models. Traditional machine learning models like linear regression, decision trees, and support vector machines typically have simpler architectures compared to neural networks. They often involve fewer parameters and computations, resulting in lower computational complexity. Also, training recurrent neural networks involves iterative computation and iterative optimization algorithms (e.g., gradient descent) that require computing gradients and updating parameters through multiple layers, which can be computationally intensive. Moreover, as continues RNNs were used as backbones in the models proposed in this thesis, they introduced further computation burden. For instance, the ODE-RNN involves numerical integration techniques like Euler's method, Runge-Kutta methods, or more sophisticated ones like adaptive step-size methods; the RNN-decay requires to store all interpolated latent states for the case of computing gradients, which can also leads to higher complexity in gradient calculation. In practice, the time taken to train a continuous RNN model is on average five to ten times longer compared to a discrete RNN model, holding all other variables constant. One trick to accelerate the training process of the proposed models might be to utilising a larger batch size. However, this must be balanced with the project's constraints on computational resources (24 GB), particularly considering the trade-off between accelerating parallel computation and the constraints imposed by limited GPU graphics memory. In terms of model inference, the batch size can only be set as one as the prediction on the next LOB snapshots or the next event arriving in the market is recursive and is dependent on the former prediction. In this sense, utilising a larger batch size cannot accelerate inference. Notably, computational complexity can be mitigated by adopting less-aggressive hyperparameters and trading off model accuracy. Alternatively, techniques like intensity free modelling of temporal point process [165] might be adopted to alleviate this high computation burden.

In terms of some model-specific limitations:

- The LOBRM presented in Chapter 4 was designed to recover deep level LOB volume information based on top level TAQ data. Although the LOBRM outperformed other conventional regression and deep learning models in this specific task, its capacity to account for the variance in high frequency predictions remained limited, as evidenced by a low coefficient of determination. The main reason might be the noisy and chaotic nature of the LOB, making the task intrinsically difficult. On the other hand, the model was 'static' in the sense that it did not consider the event-based dynamic of a real LOB, leading to its failure in conducting realistic LOB simulation.

- The sd-PNHP presented in Chapter 5 was designed to predict information concerning the next event arriving at the LOB based on historical event stream. To simplify the modelling

process, events in the LOB were categorised into only four basic types, namely ask submission, ask cancellation, bid submission and bid cancellation. In real financial exchanges, the event types are of higher variety, for instance market orders or modification of orders, not to say the more diversified types of limit orders affiliated with different execution styles. Considering this diversity in event types might lead to further improvement in model performance, especially for a more fine-grained LOB simulation. On the other hand, the proposed sd-PNHP has the potential to be applicable to a wide range of similar tasks involving multi-variate and event-state inter-dependent data, while this generality of the model was not investigated.

- The NS-ABM presented in Chapter 6 was a hybrid realistic LOB simulation framework, in which a BT with sd-PNHP and DDPM backbones was embedded to represent the market. The core idea was to improve the objectivity in LOB simulation compared with traditional ABM methodologies. Even though the majority of settings of the NS-ABM has been grounded on real data, the simulation was still partially rule-based, for instance some of the emergency settings in the simulation. How to make the simulation fully objective and realistic is still a path to be further explored. Further, the validation of a LOB simulation system has always been insufficiently researched. Currently, the validity of a LOB simulation can only be judged by its presented stochastic properties and certain responses to external actions, and different research used different sets of standards to evaluate their simulations. Even though the thesis has tried utmost best to use a comprehensive list of criteria to evaluate the validity of the simulation, the difficulty of evaluating a model without solid ground truth still remains.

7.3 Future Works

Based on the research journey of the thesis and the author's understanding of the research field, several meaningful future research directions are listed:

- Validating and improving the generality of the proposed models. As stated earlier, the generality of the proposed models in a wider range of application scenarios has not been validated. For instance, it would be interesting to see how the models perform for other asset classes that are much less liquid (e.g. sneakers auctioned on the *StockX* platform); and how the models perform in different while similar tasks (e.g. the performance of sd-PNHP in predicting medical visits). At the same time, if a large dataset is permitted, it would also be interesting to investigate whether the model performance decay with time owing to stochastic drift in the market.

- Incorporating enhanced dynamics in LOB simulation. The training data on which models were trained was too small to consider some of the structural changes or shocks in the

LOB. For instance, suppose a LOB dataset with years length is available, different models can be trained on 'normal' LOB and 'abnormal' LOB separately. The 'abnormal' LOB data can be resulted from market crushes or release of financial reports among other factors. By looking deep into the difference of the results produced by different models, it can be verified whether the models are capable of learning the structural differences in the LOB brought by external information. If so, enhanced dynamics can be incorporated in the simulation by using models trained on different data as the backbone for the BT, in order to replay a certain market dynamics existed in the history. This would provide a valuable scenario analysis for researchers and practitioners.

- Utilising more advanced deep learning techniques in LOB modelling. The research on applying deep learning techniques to traditional financial modelling tasks has just begun over the past decade, and is far from being mature. The potential of deep learning models has not been fully tapped. For instance, relational models like graph neural networks (GNNs) [166] have not yet been thoroughly investigated in the field of LOB modelling. Being a structural model composed of nodes and edges, GNNs allow information to spread within a graph derived from prior knowledge between entities. The model is naturally suitable for modelling the information flow between market participants or different markets trading a same asset, and at the same time the influence of the exchange of information on the model's prediction performance. Another possibility would be to treat the LOB directly as a visualised heatmap, or a picture. Thus, the continuous prediction or simulation of the LOB can be regarded as a video processing task. Treating the task this way, some state-of-the-art deep learning techniques widely used in the field of computer vision like CNNs [167] or Transformers [168] can be easily adapted to be applied.

- Handling time irregularities in the LOB more effectively. Even though continuous time RNN models have been adopted to handle time irregularities of the LOB in this thesis, more attempts can be explored concerning this topic. First, alternative model structures can be considered. Temporal convolutional networks (TCNs) [169] represent another promising avenue for handling time irregularities in LOB data. TCNs leverage dilated convolutions to capture long-range dependencies in time series data efficiently. By incorporating TCNs alongside continuous time RNN models, the framework can exploit the complementary strengths of both architectures to better model irregular time intervals and complex temporal patterns present in LOB data. Second, a hierarchical approach to modelling temporal irregularities can further bolster the model's resilience. By organizing data into hierarchical structures that represent different temporal scales or levels of abstraction, the model can adapt to irregularities at both local and global temporal contexts. This hierarchical framework equips the model with the flexibility to discern patterns across multiple temporal resolutions, thereby enhancing its capacity to capture the nuances of LOB data with varying time intervals. Additionally, resampling techniques offer a promising avenue for

mitigating irregularities in LOB data. Time-based resampling involves aggregating data within regular time intervals, ensuring consistency and manageability for analysis. Conversely, event-based resampling defines new time intervals based on specific market events, capturing meaningful activity even amidst irregular intervals. Rather than rule-based resampling, attention mechanism [168] can dynamically adjust the sampling rate based on the relevance and importance of different time intervals. By prioritizing informative periods while deprioritizing less relevant ones, attention-based resampling optimizes computational resources and improves the model's efficiency in handling irregular time intervals. These resampling techniques help regularize the data, reduce noise, and provide better control over the granularity of the time series, ultimately improving the performance of predictive models and facilitating more accurate insights.

- Exploring the potential of reinforcement learning in terms of LOB simulation. Reinforcement learning (RL) [170, 171] stands of substantial significance in the fast-evolving field of deep learning. The core of RL algorithms is to train an intelligent agent to learn an optimal strategy stream to maximize its profit or utility gained from the interaction with a specific dynamic environment. The environment can be a video game program, a chess game, or under the general setting of this paper, a CDA market. The focus of RL in the domain of financial microstructure has been on the trading side. To put it simply, to train a trading agent whose behavior logic is controlled by neural networks in the hope of maximizing its trading profit. In the study by Wei et al. [102], the trading problem was formulated as a Markov Decision Process (MDP) consisting of a latent representation model, RNN-based transition model, and reward model. A CNN-based auto-encoder was used to represent the LOB data with latent states. The RL agent learned to update its policy to maximize cumulative rewards over a fixed time horizon. By validating on real LOB data from the *Hong Kong Stock Exchange*, the RL-based trading strategy was shown to beat rule-based strategies like momentum strategy and classifier-based strategy. Similarly, Gašperov and Kostanjčar [172] explored the potential of utilizing RL for market making. In terms of market simulation, Karpe et al. [139] proposed a model-free approach for optimal order execution in a limit order book market using multi-agent reinforcement learning (RL). They developed and trained an RL execution agent using the Double Deep Q-Learning algorithm and evaluated it by comparing it with a market replay simulation using real market data. The paper highlighted the challenges and importance of developing interactive agent-based simulations for trading strategies and provided insights into the use of RL in realistic market simulations. While the focus of the study was still on investigating the profitability of the RL agent, instead of validating the rationale and authenticity of the simulation method. We hope for more research on RL-based LOB simulation to appear in the future.

  In conclusion, the flourishing cross-disciplinary domain of deep learning driven LOB modelling and simulation presents both immense potential and inherent challenges. It is

the aspiration of this thesis to enhance comprehension of financial markets and to serve as a catalyst for subsequent research endeavors with analogous objectives.

BIBLIOGRAPHY

[1] R. Kissell, *Algorithmic trading methods: Applications using advanced statistics, optimization, and machine learning techniques*. Academic Press, 2021.

[2] D. Friedman, "The double auction market institution: A survey," *The double auction market: Institutions, theories, and evidence*, vol. 14, pp. 3–25, 1993.

[3] M. D. Gould, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, "Limit order books," *Quantitative Finance*, vol. 13, no. 11, pp. 1709–1742, 2013.

[4] B. Biais, L. Glosten, and C. Spatt, "Market microstructure: A survey of microfoundations, empirical results, and policy implications," *Journal of Financial Markets*, vol. 8, no. 2, pp. 217–264, 2005.

[5] B. Biais, P. Hillion, and C. Spatt, "An empirical analysis of the limit order book and the order flow in the Paris Bourse," *Journal of Finance*, vol. 50, no. 5, pp. 1655–1689, 1995.

[6] C. A. Parlour, "Price dynamics in limit order markets," *The Review of Financial Studies*, vol. 11, no. 4, pp. 789–816, 1998.

[7] T. Foucault, O. Kadan, and E. Kandel, "Limit order book as a market for liquidity," *The review of financial studies*, vol. 18, no. 4, pp. 1171–1217, 2005.

[8] R. Cont, S. Stoikov, and R. Talreja, "A stochastic model for order book dynamics," *Operations research*, vol. 58, no. 3, pp. 549–563, 2010.

[9] A. Cartea, R. Donnelly, and S. Jaimungal, "Enhancing trading strategies with order book signals," *Applied Mathematical Finance*, vol. 25, no. 1, pp. 1–35, 2018.

[10] F. McGroarty, A. Booth, E. Gerding, and V. R. Chinthalapati, "High frequency trading strategies, market fragility and price spikes: an agent based model perspective," *Annals of Operations Research*, vol. 282, no. 1, pp. 217–244, 2019.

[11] R. Cont and A. De Larrard, "Price dynamics in a markovian limit order market," *SIAM Journal on Financial Mathematics*, vol. 4, no. 1, pp. 1–25, 2013.

[12] J. Li, X. Wang, Y. Lin, A. Sinha, and M. Wellman, "Generating realistic stock market order streams," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 727–734.

[13] J. Cartlidge, N. P. Smart, and Y. Talibi Alaoui, "Mpc joins the dark side," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019, pp. 148–159.

[14] Z. Zhang, S. Zohren, and S. Roberts, "Deeplob: Deep convolutional neural networks for limit order books," *IEEE Transactions on Signal Processing*, vol. 67, no. 11, pp. 3001–3012, 2019.

[15] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," *arXiv preprint arXiv:1806.07366*, 2018.

[16] H. Mei and J. M. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] E. Bacry, S. Delattre, M. Hoffmann, and J.-F. Muzy, "Modelling microstructure noise with mutually exciting point processes," *Quantitative finance*, vol. 13, no. 1, pp. 65–77, 2013.

[18] J. Large, "Measuring the resiliency of an electronic limit order book," *Journal of Financial Markets*, vol. 10, no. 1, pp. 1–25, 2007.

[19] F. Gonzalez and M. Schervish, "Instantaneous order impact and high-frequency strategy optimization in limit order books," *Market Microstructure and Liquidity*, vol. 3, no. 02, p. 1850001, 2017.

[20] I. Muni Toke and N. Yoshida, "Modelling intensities of order flows in a limit order book," *Quantitative Finance*, vol. 17, no. 5, pp. 683–701, 2017.

[21] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[22] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[23] M. Potters and J.-P. Bouchaud, "More statistical properties of order books and price impact," *Physica A: Statistical Mechanics and its Applications*, vol. 324, no. 1-2, pp. 133–140, 2003.

[24] E. Smith, J. D. Farmer, L. Gillemot, and S. Krishnamurthy, "Statistical theory of the continuous double auction," *Quantitative finance*, vol. 3, no. 6, p. 481, 2003.

[25] J.-P. Bouchaud, M. Mézard, and M. Potters, "Statistical properties of stock order books: empirical results and models," *Quantitative finance*, vol. 2, no. 4, p. 251, 2002.

[26] D. Challet and R. Stinchcombe, "Analyzing and modeling 1+ 1d markets," *Physica A: Statistical Mechanics and its Applications*, vol. 300, no. 1-2, pp. 285–299, 2001.

[27] G.-H. Mu, W. Chen, J. Kertész, and W.-X. Zhou, "Preferred numbers and the distributions of trade sizes and trading volumes in the chinese stock market," *The European Physical Journal B*, vol. 68, pp. 145–152, 2009.

[28] S. Maslov and M. Mills, "Price fluctuations from the order book perspective—empirical facts and a simple model," *Physica A: Statistical Mechanics and its Applications*, vol. 299, no. 1-2, pp. 234–246, 2001.

[29] L. R. Glosten, "Is the electronic open limit order book inevitable?" *The Journal of Finance*, vol. 49, no. 4, pp. 1127–1161, 1994.

[30] C. M. Lee, "Market integration and price execution for nyse-listed securities," *The Journal of Finance*, vol. 48, no. 3, pp. 1009–1038, 1993.

[31] P. Sandås, "Adverse selection and competitive market making: Empirical evidence from a limit order market," *The review of financial studies*, vol. 14, no. 3, pp. 705–734, 2001.

[32] A. Ellul, C. W. Holden, P. Jain, and R. Jennings, "Determinants of order choice on the new york stock exchange," *manuscript, Indiana University*, 2003.

[33] A. D. Hall and N. Hautsch, "Order aggressiveness and order book dynamics," *Empirical Economics*, vol. 30, pp. 973–1005, 2006.

[34] C. Cao, O. Hansch, and X. Wang, "Order placement strategies in a pure limit order book market," *Journal of Financial Research*, vol. 31, no. 2, pp. 113–140, 2008.

[35] F. Lillo and J. D. Farmer, "The long memory of the efficient market," *Studies in nonlinear dynamics & econometrics*, vol. 8, no. 3, 2004.

[36] G.-F. Gu and W.-X. Zhou, "Emergence of long memory in stock volatility from a modified mike-farmer model," *EPL (Europhysics Letters)*, vol. 86, no. 4, p. 48002, 2009.

[37] R. Cont, "Volatility clustering in financial markets: empirical facts and agent-based models," in *Long memory in economics*. Springer, 2007, pp. 289–309.

[38] E. F. Fama, "The behavior of stock-market prices," *The journal of Business*, vol. 38, no. 1, pp. 34–105, 1965.

[39] P. Gopikrishnan, M. Meyer, L. N. Amaral, and H. E. Stanley, "Inverse cubic law for the distribution of stock price variations," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 3, no. 2, pp. 139–140, 1998.

[40] V. Plerou and H. E. Stanley, "Stock return distributions: tests of scaling and universality from three distinct stock markets," *Physical Review E*, vol. 77, no. 3, p. 037101, 2008.

[41] G.-F. Gu, W. Chen, and W.-X. Zhou, "Empirical distributions of chinese stock returns at different microscopic timescales," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 2-3, pp. 495–502, 2008.

[42] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quantitative finance*, vol. 1, no. 2, p. 223, 2001.

[43] P. Gopikrishnan, V. Plerou, L. A. N. Amaral, M. Meyer, and H. E. Stanley, "Scaling of the distribution of fluctuations of financial market indices," *Physical Review E*, vol. 60, no. 5, p. 5305, 1999.

[44] H. E. Stanley, V. Plerou, and X. Gabaix, "A statistical physics view of financial fluctuations: Evidence for scaling and universality," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 15, pp. 3967–3981, 2008.

[45] R. Cont, "Long range dependence in financial markets," in *Fractals in engineering: New trends in theory and applications*. Springer, 2005, pp. 159–179.

[46] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[47] R. Almgren and N. Chriss, "Optimal execution of portfolio transactions," *Journal of Risk*, vol. 3, pp. 5–40, 2001.

[48] J.-P. Bouchaud, J. D. Farmer, and F. Lillo, "How markets slowly digest changes in supply and demand," in *Handbook of financial markets: dynamics and evolution*. Elsevier, 2009, pp. 57–160.

[49] J. Hasbrouck, "Measuring the information content of stock trades," *The Journal of Finance*, vol. 46, no. 1, pp. 179–207, 1991.

[50] F. Lillo, J. D. Farmer, and R. N. Mantegna, "Master curve for price-impact function," *Nature*, vol. 421, no. 6919, pp. 129–130, 2003.

[51] C. Hopman, "Do supply and demand drive stock prices?" *Quantitative Finance*, vol. 7, no. 1, pp. 37–53, 2007.

[52] M. G. Daniels, J. D. Farmer, G. Iori, and E. Smith, "How storing supply and demand affects price diffusion," *arXiv preprint cond-mat/0112422*, 2002.

[53] J. D. Farmer and F. Lillo, "On the origin of power-law tails in price fluctuations," *Quantitative Finance*, vol. 4, no. 1, p. C7, 2004.

[54] A. Kempf and O. Korn, "Market depth and order size," *Journal of Financial Markets*, vol. 2, no. 1, pp. 29–48, 1999.

[55] X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley, "Institutional investors and stock market volatility," *The Quarterly Journal of Economics*, vol. 121, no. 2, pp. 461–504, 2006.

[56] M. D. Evans and R. K. Lyons, "Order flow and exchange rate dynamics," *Journal of political economy*, vol. 110, no. 1, pp. 170–180, 2002.

[57] R. Cont, A. Kukanov, and S. Stoikov, "The price impact of order book events," *Journal of financial econometrics*, vol. 12, no. 1, pp. 47–88, 2014.

[58] U. Securities, E. Commission *et al.*, "Part iii: Concept release on equity market structure; proposed rule, 17 cfr part 242," *Federal Register*, vol. 75, no. 13, pp. 3594–3614, 2010.

[59] K. Z. Zaharudin, M. R. Young, and W.-H. Hsu, "High-frequency trading: Definition, implications, and controversies," *Journal of Economic Surveys*, vol. 36, no. 1, pp. 75–107, 2022.

[60] J. Brogaard, T. Hendershott, and R. Riordan, "High-frequency trading and price discovery," *The Review of Financial Studies*, vol. 27, no. 8, pp. 2267–2306, 2014.

[61] J. Conrad, S. Wahal, and J. Xiang, "High-frequency quoting, trading, and the efficiency of prices," *Journal of Financial Economics*, vol. 116, no. 2, pp. 271–291, 2015.

[62] K. A. Froot, D. S. Scharfstein, and J. C. Stein, "Herd on the street: Informational inefficiencies in a market with short-term speculation," *The Journal of Finance*, vol. 47, no. 4, pp. 1461–1484, 1992.

[63] R. A. Jarrow and P. Protter, "A dysfunctional role of high frequency trading in electronic markets," *International Journal of Theoretical and Applied Finance*, vol. 15, no. 03, p. 1250022, 2012.

[64] A. Carrion, "Very fast money: High-frequency trading on the nasdaq," *Journal of Financial Markets*, vol. 16, no. 4, pp. 680–711, 2013.

[65] V. Manahov, R. Hudson, and B. Gebka, "Does high frequency trading affect technical analysis and market efficiency? and if so, how?" *Journal of International Financial Markets, Institutions and Money*, vol. 28, pp. 131–157, 2014.

[66] A. Cartea, R. Payne, J. Penalva, and M. Tapia, "Ultra-fast activity and intraday market quality," *Journal of Banking & Finance*, vol. 99, pp. 157–181, 2019.

[67] A. Shkilko and K. Sokolov, "Every cloud has a silver lining: Fast trading, microwave connectivity, and trading costs," *The Journal of Finance*, vol. 75, no. 6, pp. 2899–2927, 2020.

[68] X. Wang, C. Hoang, Y. Vorobeychik, and M. P. Wellman, "Spoofing the limit order book: A strategic agent-based analysis," *Games*, vol. 12, no. 2, p. 46, 2021.

[69] M. Zare, O. Naghshineh Arjmand, E. Salavati, and A. Mohammadpour, "An agent-based model for limit order book: Estimation and simulation," *International Journal of Finance & Economics*, vol. 26, no. 1, pp. 1112–1121, 2021.

[70] S. Jacob Leal, M. Napoletano, A. Roventini, and G. Fagiolo, "Rock around the clock: An agent-based model of low-and high-frequency trading," *Journal of Evolutionary Economics*, vol. 26, pp. 49–76, 2016.

[71] S. J. Leal and M. Napoletano, "Market stability vs. market resilience: Regulatory policies experiments in an agent-based model with low-and high-frequency trading," *Journal of Economic Behavior & Organization*, vol. 157, pp. 15–41, 2019.

[72] E. Wah and M. P. Wellman, "Latency arbitrage in fragmented markets: A strategic agent-based analysis," *Algorithmic Finance*, vol. 5, no. 3-4, pp. 69–93, 2016.

[73] T. Preis, S. Golke, W. Paul, and J. J. Schneider, "Statistical analysis of financial returns for a multiagent order book model of asset trading," *Physical Review E*, vol. 76, no. 1, p. 016108, 2007.

[74] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[75] R. Cont and M. S. Muller, "A stochastic partial differential equation model for limit order book dynamics," *SIAM Journal on Financial Mathematics*, vol. 12, no. 2, pp. 744–787, 2021.

[76] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.

[77] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[78] R. L. Stratonovich, "Application of the markov processes theory to optimal filtering," *Radio Engineering and Electronic Physics*, vol. 5, pp. 1–19, 1960.

[79] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[80] A. Cauchy *et al.*, "Méthode générale pour la résolution des systemes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, no. 1847, pp. 536–538, 1847.

[81] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. Pmlr, 2013, pp. 1310–1318.

[82] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[83] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[84] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[85] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 2013, pp. 6645–6649.

[86] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[87] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[88] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[89] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[90] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[91] Z. C. Lipton, D. Kale, and R. Wetzel, "Directly modeling missing data in sequences with rnns: Improved classification of clinical time series," in *Machine learning for healthcare conference*. PMLR, 2016, pp. 253–270.

[92] L. S. Pontryagin, *The Mathematical Theory of Optimal Processes*. Interscience Publishers, 1962.

[93] Y. Rubanova, R. T. Chen, and D. Duvenaud, "Latent odes for irregularly-sampled time series," *arXiv preprint arXiv:1907.03907*, 2019.

[94] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[95] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[96] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[97] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," in *5th International Conference on Learning Representations, ICLR 2017*, 2019.

[98] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[99] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022.

[100] S. Hu, W. Yu, Z. Chen, and S. Wang, "Medical image reconstruction using generative adversarial network for alzheimer disease assessment with class-imbalance problem," in *2020 IEEE 6th international conference on computer and communications (ICCC)*. IEEE, 2020, pp. 1323–1327.

[101] P. Nagy, J.-P. Calliess, and S. Zohren, "Asynchronous deep double duelling q-learning for trading-signal execution in limit order book markets," *arXiv preprint arXiv:2301.08688*, 2023.

[102] H. Wei, Y. Wang, L. Mangu, and K. Decker, "Model-based reinforcement learning for predictions and control for limit order books," *arXiv preprint arXiv:1910.03743*, 2019.

[103] J. Sirignano and R. Cont, "Universal features of price formation in financial markets: perspectives from deep learning," *Quantitative Finance*, vol. 19, no. 9, pp. 1449–1459, 2019.

[104] A. Ntakaris, G. Mirone, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Feature engineering for mid-price prediction with deep learning," *Ieee Access*, vol. 7, pp. 82 390–82 412, 2019.

[105] A. Arroyo, A. Cartea, F. Moreno-Pino, and S. Zohren, "Deep attentive survival analysis in limit order books: Estimating fill probabilities with convolutional-transformers," *arXiv preprint arXiv:2306.05479*, 2023.

[106] C. Maglaras, C. C. Moallemi, and M. Wang, "A deep learning approach to estimating fill probabilities in a limit order book," *Quantitative Finance*, vol. 22, no. 11, pp. 1989–2003, 2022.

[107] W. B. Arthur, J. H. Holland, B. LeBaron, R. Palmer, and P. Tayler, "Asset pricing under endogenous expectations in an artificial stock market," in *The economy as an evolving complex system II*. CRC Press, 2018, pp. 15–44.

[108] S. Stotter, J. Cartlidge, and D. Cliff, "Behavioural investigations of financial trading agents using exchange portal (expo)," in *Transactions on Computational Collective Intelligence XVII*. Springer, 2014, pp. 22–45.

[109] D. Cliff, "An open-source limit-order-book exchange for teaching and research," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1853–1860.

[110] F. Prenzel, R. Cont, M. Cucuringu, and J. Kochems, "Dynamic calibration of order flow models with generative adversarial networks," in *Proceedings of the Third ACM International Conference on AI in Finance*, 2022, pp. 446–453.

[111] A. Coletta, A. Moulin, S. Vyetrenko, and T. Balch, "Learning to simulate realistic limit order book markets from data as a world agent," in *Proceedings of the Third ACM International Conference on AI in Finance*, 2022, pp. 428–436.

[112] X. Lu and F. Abergel, "High-dimensional hawkes processes for limit order books: modelling, empirical analysis and numerical calibration," *Quantitative Finance*, vol. 18, no. 2, pp. 249–264, 2018.

[113] S. Xiao, J. Yan, X. Yang, H. Zha, and S. Chu, "Modeling the intensity function of point process via recurrent neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[114] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, "Transformer hawkes process," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 692–11 702.

[115] C. Cao, O. Hansch, and X. Wang, "The information content of an open limit-order book," *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, vol. 29, no. 1, pp. 16–41, 2009.

[116] I. Muni Toke, "The order book as a queueing system: average depth and influence of the size of limit orders," *Quantitative Finance*, vol. 15, no. 5, pp. 795–808, 2015.

[117] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes. Volume I: elementary theory and methods*. New York, NY: Springer, 2003.

[118] Z. Shi and J. Cartlidge, "The limit order book recreation model (lobrm): An extended analysis," in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part IV 21*. Springer, 2021, pp. 204–220.

[119] Z. Shi, Y. Chen, and J. Cartlidge, "The lob recreation model: Predicting the limit order book from taq history using an ordinary differential equation recurrent neural network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 548–556.

[120] J. Blanchet, X. Chen, and Y. Pei, "Unraveling limit order books using just bid/ask prices," *Preprint. Available at http://www. columbia. edu/ jb2814/papers/LOB_v1. pdf*, 2017.

[121] K. Dayri and M. Rosenbaum, "Large tick assets: implicit spread and optimal tick size," *Market Microstructure and Liquidity*, vol. 1, no. 01, p. 1550003, 2015.

[122] S. M. Krause, E. Jungblut, and T. Guhr, "Two price regimes in limit order books: liquidity cushion and fragmented distant field," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2022, no. 2, p. 023401, 2022.

[123] U. Horst and D. Kreher, "A weak law of large numbers for a limit order book model with fully state dependent order dynamics," *SIAM Journal on Financial Mathematics*, vol. 8, no. 1, pp. 314–343, 2017.

[124] A. Ntakaris, M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods," *Journal of Forecasting*, vol. 37, no. 8, pp. 852–866, 2018.

[125] Z. Shi and J. Cartlidge, "State dependent parallel neural hawkes process for limit order book event stream prediction and simulation," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1607–1615.

[126] Q. Zhang, A. Lipani, O. Kirnap, and E. Yilmaz, "Self-attentive Hawkes process," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. MLResearchPress, 13–18 Jul 2020, pp. 11 183–11 193.

[127] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1555–1564.

[128] Y. Gu, "Attentive neural point processes for event forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 7592–7600.

[129] Y. Zhang, K. Sharma, and Y. Liu, "Vigdet: Knowledge informed neural temporal point process for coordination detection on social media," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3218–3231, 2021.

[130] Y. Ogata, "On lewis' simulation method for point processes," *IEEE transactions on information theory*, vol. 27, no. 1, pp. 23–31, 1981.

[131] M. Morariu-Patrichi and M. S. Pakkanen, "State-dependent hawkes processes and their application to limit order book modelling," *Quantitative Finance*, vol. 22, no. 3, pp. 563–583, 2022.

[132] I. M. Sobol, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates," *Mathematics and computers in simulation*, vol. 55, no. 1-3, pp. 271–280, 2001.

[133] J.-P. Bouchaud and M. Potters, *Theory of financial risk and derivative pricing: from statistical physics to risk management*. Cambridge university press, 2003.

[134] Z. Shi and J. Cartlidge, "Neural stochastic agent-based limit order book simulation: A hybrid methodology," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 2481–2483.

[135] L. Feng, B. Li, B. Podobnik, T. Preis, and H. E. Stanley, "Linking agent-based models and stochastic models of financial markets," *Proceedings of the National Academy of Sciences*, vol. 109, no. 22, pp. 8388–8393, 2012.

[136] E. Panayi and G. W. Peters, "Stochastic simulation framework for the limit order book using liquidity-motivated agents," *International Journal of Financial Engineering*, vol. 2, no. 02, p. 1550013, 2015.

[137] P. Kumar, "Deep Hawkes process for high-frequency market making," arXiv:2109.15110, 2021, https://doi.org/10.48550/arXiv.2109.15110.

[138] D. Byrd, M. Hybinette, and T. H. Balch, "Abides: Towards high-fidelity multi-agent market simulation," in *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 2020, pp. 11–22.

[139] M. Karpe, J. Fang, Z. Ma, and C. Wang, "Multi-agent reinforcement learning in a realistic limit order book market simulation," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–7.

[140] C. Yagemann, S. P. Chung, E. Uzun, S. Ragam, B. Saltaformaggio, and W. Lee, "On the feasibility of automating stock market manipulation," in *Annual Computer Security Applications Conference*, 2020, pp. 277–290.

[141] D. Byrd, "Explaining agent-based financial market simulation," *arXiv preprint arXiv:1909.11650*, 2019.

[142] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, vol. 27, 2014.

[143] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[144] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of dna nucleotides," *Physical review e*, vol. 49, no. 2, p. 1685, 1994.

[145] E. F. Fama, "Efficient capital markets: Ii," *The journal of finance*, vol. 46, no. 5, pp. 1575–1617, 1991.

[146] S. Vyetrenko, D. Byrd, N. Petosa, M. Mahfouz, D. Dervovic, M. Veloso, and T. Balch, "Get real: Realism metrics for robust limit order book market simulations," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–8.

[147] F. Abergel, M. Anane, A. Chakraborti, A. Jedidi, and I. M. Toke, *Limit order books*. Cambridge University Press, 2016.

[148] O. Brandouy, A. Corelli, I. Veryzhenko, and R. Waldeck, "A re-examination of the "zero is enough" hypothesis in the emergence of financial stylized facts," *Journal of Economic Interaction and Coordination*, vol. 7, pp. 223–248, 2012.

[149] J.-P. Bouchaud and M. Potters, "More stylized facts of financial markets: leverage effect and downside correlations," *Physica A: Statistical Mechanics and its Applications*, vol. 299, no. 1-2, pp. 60–70, 2001.

[150] J.-M. Dufour, R. Garcia, and A. Taamouti, "Measuring high-frequency causality between returns, realized volatility, and implied volatility," *Journal of Financial Econometrics*, vol. 10, no. 1, pp. 124–163, 2012.

[151] J.-J. Chen, B. Zheng, and L. Tan, "Agent-based model with asymmetric trading and herding for complex financial systems," *PloS one*, vol. 8, no. 11, p. e79531, 2013.

[152] R. T. Zhou and R. N. Lai, "Herding and information based trading," *Journal of Empirical Finance*, vol. 16, no. 3, pp. 388–393, 2009.

[153] N. E. Boyd, B. Büyükşahin, M. S. Haigh, and J. H. Harris, "The prevalence, sources, and effects of herding," *Journal of Futures Markets*, vol. 36, no. 7, pp. 671–694, 2016.

[154] Q. Chen, X. Hua, and Y. Jiang, "Contrarian strategy and herding behaviour in the chinese stock market," *The European Journal of Finance*, vol. 24, no. 16, pp. 1552–1568, 2018.

[155] S. Bikhchandani and S. Sharma, "Herd behavior in financial markets," *IMF Staff papers*, vol. 47, no. 3, pp. 279–310, 2000.

[156] S. Schulmeister, "Profitability of technical stock trading: Has it moved from daily to intraday data?" *Review of Financial Economics*, vol. 18, no. 4, pp. 190–201, 2009.

[157] E. Boehmer, D. Li, and G. Saar, "The competitive landscape of high-frequency trading firms," *The Review of Financial Studies*, vol. 31, no. 6, pp. 2227–2276, 2018.

[158] A. S. Serrano, "High-frequency trading and systemic risk: A structured review of findings and policies," *Review of Economics*, vol. 71, no. 3, pp. 169–195, 2020.

[159] E. J. Lee, "High frequency trading in the korean index futures market," *Journal of Futures Markets*, vol. 35, no. 1, pp. 31–51, 2015.

[160] B. Biais, T. Foucault *et al.*, "Hft and market quality," *Bankers, Markets & Investors*, vol. 128, no. 1, pp. 5–19, 2014.

[161] S. Nawn and A. Banerjee, "Do the limit orders of proprietary and agency algorithmic traders discover or obscure security prices?" *Journal of Empirical Finance*, vol. 53, pp. 109–125, 2019.

[162] N. Bershova and D. Rakhlin, "The non-linear market impact of large trades: Evidence from buy-side order flow," *Quantitative finance*, vol. 13, no. 11, pp. 1759–1778, 2013.

[163] R. F. Almgren, "Optimal execution with nonlinear impact functions and trading-enhanced risk," *Applied mathematical finance*, vol. 10, no. 1, pp. 1–18, 2003.

[164] F. Capponi and R. Cont, "Trade duration, volatility and market impact," *Volatility and Market Impact (March 14, 2019)*, 2019.

[165] O. Shchur, M. Biloš, and S. Günnemann, "Intensity-free learning of temporal point processes," *arXiv preprint arXiv:1909.12127*, 2019.

[166] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[167] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[168] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[169] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[170] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[171] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[172] B. Gašperov and Z. Kostanjčar, "Deep reinforcement learning for market making under a hawkes process-based limit order book model," *IEEE control systems letters*, vol. 6, pp. 2485–2490, 2022.