Master’s Thesis

AN ANALYSIS OF OPTIMAL TRADING EXECUTION USING Q-LEARNING IN THE PRESENCE OF MARKET IMPACT OF LIMIT AND MARKET ORDERS

Chair of Finance – Professor Dr. Erik Theissen
Advisor: Professor Dr. Erik Theissen
Student: Nguyen Thi Hoa
Address: Ulmenweg 55, House 4, Room 318, 68167 Mannheim
Student ID: 1712894
University of Mannheim, 04 September 2021

Table of Contents

List of Tables

List of Figures

I. Introduction

The burgeoning assets of institutional investors such as hedge funds, mutual funds and pension funds, and the technological innovation over the past decades have driven tremendous growth in equity trading. Nowadays an increasingly large number of assets are traded electronically. Electronic limit order book (LOB) contains both bid and ask orders created by market participants and is maintained by a trading venue. The increasing availability of LOB data provides investors with ample opportunities in measurement and management of trading costs which are surprisingly large for institutional investors and have eaten up a large portion of their profits of their strategies. Over the last decades, there have been considerable interests in optimized trade execution in both finance and computer science literature which is aimed to minimize trading costs. However, the empirical literature of this execution cost control problem still has many limitations, the crucial of which is the omission of price impact of both limit order and market order in the model that trains the autonomous agent to act optimally to minimize the trading costs since the price impact constitutes a large proportion of trading costs. In this thesis, I make an attempt to solve this execution cost control problem with the application of reinforcement learning, particularly Q-learning, design an action space of limit orders and market orders of various volumes for Q-learning, and include price impact of limit order and market order into Q-learning algorithm. Q-learning is a model-free approach that makes no assumptions about a model of environment and is a simple way to train agents to act optimally in controlled Markovian domains. The central result of this study is that Q-learning policies in the presence of price impact of limit order and market order outperform two benchmark strategies, namely Submit and Leave and TWAP strategy in most of the time.

In the optimal trading execution problem, it is a common practice that a large parent order is split into child orders which are executed subsequently. In the literature, child orders are usually executed restrictively, for instance, via market orders (e.g. Bersimas and Lo, 1998; Almgren and Chriss, 2001), via limit orders (e.g. Nevmyvaka, et al., 2006) or via both market orders and limit orders but the order volume is restricted to the Time-Weighted-Average-Price schedule (e.g. Vyetrenko and Xu, 2019) whereas in practice, optimal execution strategies seek the mixture of both limit and market orders of various volumes. This thesis accordingly contributes to this stream of research by expanding the action space of the agent in Q-learning algorithm with market orders and limit orders of a variety of volumes.

Another contribution of this thesis is to propose a novel method to include the price impact of both limit and market orders into Q-learning so that the agent is trained in the presence of adverse impacts caused by both limit and market orders. Market orders target immediate consumption of available liquidity but incur the bid-ask spread while limit orders incur no spread cost but bear the risk of being unfilled. Placing a large order or consecutive small orders regardless of order type may reveal trading intention and therefore adversely affect stock prices (Engle and Patton, 2004). While trades have been rigorously found in the literature to incur an impact on the market, limit order placement may also create adverse price movement. (Hautsch and Huang, 2012) argue that the magnitude and direction of quote adjustments followed by limit orders strongly depend on the order's relative size, its aggressiveness and the prevailing depth of the order book. In this thesis, I conduct an empirical analysis of mid-price changes caused by changes in order book and then estimate the impact of placing limit and market orders as expected price change due to changes in order book caused by order placement. This analysis shows that mid-price change is significantly and statistically triggered by order flow imbalance, trade imbalance and order imbalance at more than 99% confidence level and most importantly preserves the intuition that trades cause a higher price impact than limit orders and larger orders incur higher costs than small orders. The results are robust across half of trading hours and R-squared is 40.7% during the end of trading day on average for stock CATY. Then I incorporate this estimated price impact into Q-learning algorithm as temporary price impact due to an order placement of the previous time step. The idea is that the quote of the next immediate time step is adversely affected by the order placement of the previous time step. The inclusion of price impact into Q-learning fits the intuition that placement of consecutive orders triggers a higher penalty than distant order placement does. This study accounts for temporary effects of orders without considering permanent price impacts since this study stands from the viewpoint of passive investors and market impact of trades and limit orders placed by passive investors are empirically found to be transient.

To my knowledge, most of the research in this domain fails to clarify how to select relevant state space for Q-learning. In this research, I make an attempt to specify state space for Q-learning by examining the factors in order book that determine the observed optimal action of the agent using linear probability model. The observed optimal policy is the optimal policy given that the whole episode is observed, which is not readily available but can be indicated based on the observed prices and the volume filled. The findings show that among all other features, time step, moving difference (difference between current price and moving average), trading imbalance, trading volume, order imbalance and spread are key factors that decide the optimal actions. I first split my dataset into training set and test set. After determining state space, I train the agent via Q-learning with market impacts on the training set and gauge performance of the learned strategies against two popular benchmarks often used by researchers and practitioners, namely Submit and Leave and Time-Weighted-Average-Price (TWAP), on the test dataset. I find that policies learned from Q-learning outperform both Submit and Leave and TWAP strategies in both average accumulative returns and modified Sharpe ratio.

Q-learning and especially deep Q-learning usually require a large number of observations for the estimates to asymptotically converge in order to achieve the desirable results. While the large volume of data is not always available, even if it is, the tradeoff between using historical data and simulated data is still a nuisance to researchers and practitioners. Historical data reflects the true past, but not necessarily the future and the more distant the data is, the less relevant it may be to cover the future scenarios of the market. On the other hand, simulated data is cheaply generated but may also distort the true market data due to modeling error. Despite tremendous research progress over the last decades, simulation of limit order book still has been a challenge for researchers until now, especially in explaining empirical features of limit order book and modeling the interactions of multiple agents in the market. Thus, in this study, I train the agent based on both historical and simulated data and compare the performance of strategies on test dataset learned from historical data and simulated data. I build a limit order book simulator based on zero-intelligence model by Abergel and Jedidi (2013) which is very popular approach in the literature. Although a large volume of simulated data allows for convergence of many state-action Qs, the strategies learned from simulated data perform worse than the strategies learned from historical data since the simulated data generated from zero-intelligence framework by Abergel and Jedidi (2013) consistently underestimate spread which is a key factor that represents state of the order book. This is a warning for many studies that are only based on simulated data without comparing the performance with historical data such as Karpe et al. (2020). Finally, I examine the performance of Q-learning on stocks with different liquidity levels and find that Q-learning policies robustly surpass Submit and Leave and TWAP strategies in both expected returns and modified Sharpe ratio, and Q-learning policies learned from historical data outperform those learned from simulated data, for most of the stocks.

The structure of this thesis is as follows. Literature review is presented in section 2. Section 3 describes the datasets and summary statistics. Section 4 explains methodologies and empirical results based on historical data. LOB simulator and empirical results based on simulated data are discussed in section 5. Section 6 shows the robustness test for other stocks with different liquidity levels. Section 7 concludes the thesis with a summary of findings and future work.

II. Literature Review

2.1. Optimal Trading Execution

A variety of studies on optimal trading execution has been done since Bertsimas and Lo (1998). Bertsimas and Lo (1998) nicely depict the overall picture of execution-cost control problem using stochastic dynamic programming. Almgren and Chriss (2001) extend the work of Bertsimas and Lo (1998) by considering both volatility risk and expected transaction costs arising from permanent and temporary market impacts. They argue that this framework of risk in execution yields important results which are consistent with intuition. For instance, for illiquid securities, a trader would execute less rapidly than for liquid securities. Almgren (2003) examines the optimal execution strategies with nonlinear impact function, particularly a power law function of trading rate and investigates the problem under the scenario of increased uncertainty of realized price by demanding rapid execution. However, three of the papers use dynamic programming framework which assumes a perfect model of the environment as a Markov decision process (MDP) and is computationally expensive. Dynamic programming is important theoretically but due to its strong assumption and enormous computational expense, it is not so attractive empirically. The urgency of trading can be addressed through temporary price impact and discount rate within Q-learning framework. If the asset is illiquid, the empirical price impact would be large, thereby executing rapidly considered suboptimal. Discount rate accounts for time value of money, which can be employed as a hyperparameter to reflect the urgency of trading. The higher discount rate is, the more valuable the immediate cashflow is to the investors and the more urgent the trading is. Other forms of risk aversion for trading urgency can be easily plugged into the algorithm. Agliardi and Gencay (2017) also consider the execution control problem but with limit orders only. These papers only use market orders or limit orders as the optimal strategies while in practice, the mix of both limit orders and market orders is optimal. Market orders guarantee execution but are more expensive due to spread cost and higher price impact (Alfonsi et al., 2010; Gueant et al., 2012), whereas limit orders may not be filled but are less expensive due to no spread cost and probably a rebate earned for providing liquidity (Cartea and Jaimungal, 2013). Cartea and Jaimungal (2015) also show theoretically how to include limit orders, market orders and different volume levels into the action space of dynamic programming problem.

Related to this body of research, there are a variety of theoretical papers that extend the work of Almgren and Chriss (2001) to other related problems. Particularly, Vaes and Hauser (2018) try to derive the optimal execution strategy with an uncertain volume which is typically an issue in power market. Another example is Fruth et al. (2019) who discuss the optimal execution trading with stochastic liquidity. Stochastic liquidity is one of the major issues in trading. Some part of the liquidity is driven by the deterministic changes in order books, but other parts may be driven by exogenous information such as news release. This stochastic aspect could lead the shock of the order book. This state of the order book may be rarely visited, which renders the agent to act sub-optimally under this state of environment. This issue can be solved empirically by training the agent with simulated data simulating both deterministic and stochastic depth of the book. That said, accurate simulation of the order book is a challenging task that demands further research. Forsyth et al. (2012) and Huitema (2014) among others investigate the portfolio execution strategies to maximize expected exponential utility. They employ dynamic programming to derive a Hamilton-Jacobi-Bellman (HJB) equation.

Related to the optimal execution problem is the problem of market making and optimal trading. Theoretical research in optimal market making includes Avellaneda and Stoikov (2008) and Guibaud and Pham (2013) who study optimal strategies for market maker who provides liquidity by submitting bid and ask quotes. Jaimungal et al. (2013), Gueant et al. (2015) and among others investigate the optimal trading strategy with accelerated share repurchases. Cont and Kukanov (2017) address the optimal order placement problem with limit orders and market orders submitted across various trading venues.

2.2. Application of Reinforcement learning – Q-learning to Optimal Trading Execution

As mentioned earlier, dynamic programming framework is theoretically important but empirically unattractive due to its strong assumption of the perfect world and its highly computational expense. Reinforcement learning is a model-free approach that solves problems in Markovian world. Reinforcement learning has been applied in notable financial problems including optimal execution (e.g. Nevmyvaka et al., 2006; Vyetrenko and Xu, 2019), market making (e.g. Shelton, 2001; Spooner et al., 2018), trading (e.g. Moody et al., 1998; Schvartzman and Wellman, 2009), foreign exchange trading (e.g. Dempster and Leemans, 2006). Reinforcement learning has been progressively researched and in combination with deep learning techniques, have achieved significant breakthroughs over the last few years, especially in games (e.g. Mnih et al., 2015), robotics (e.g. Gandhi et al., 2017), to name a few.

In optimal trading execution, various studies have been performed since Nevmyvaka et al. (2006). Nevmyvaka et al. (2006) nicely introduce the application of reinforcement learning, particularly Q-learning by Watkins (1989), to optimized trade execution without the presence of market impacts. Hendricks and Wilcox (2014) extend application of reinforcement to optimal trading execution using Almgren-Chriss framework. The authors assume that child orders only can be submitted with market orders without any market impact. Vyetrenko and Xu (2019) also investigate the problem in the presence of market impacts from market orders using simulated data. However, in their model, the agent only can choose an equally fixed size of order in each micro-interval of the trading period with either market or limit order. They also do not evaluate the performance of their model relative to benchmark strategies.

Many papers have been incorporating Deep Q-learning into solutions for optimal execution since it was introduced in 2013. Deep Q-learning is the combination of Q-learning and deep learning, which is useful when important features of state space are difficult to be handcrafted or state spaces are large[1]. Deep Q-learning is usually employed with experience replay to maintain the independent and identical distribution assumption of function approximation. Ning et al. (2018) employ Double Q-learning to solve this control problem with risk-adjusted reward function. They only consider execution of market orders and account for transaction costs or risk with quadratic penalty on number of shares executed every second. Apart from the penalty term, their reward is the difference between price as of next second and price as of now, scaled by remaining inventory. This reward function is not very intuitive in the sense that usually reward function should be the difference between cashflow from the current trade and the cashflow from the benchmark strategies. The reward function is very important to train the agent to act towards the desirable strategies. They claim that their strategy mostly outperforms TWAP strategy but their evaluation metrics may not reflect the accurate price impacts. Lin and Beling (2020) make a similar attempt, but instead of using Double Q-Network, they employ Deep Q-Network with more carefully designed reward function to beat TWAP strategy. Even though they claim their model beats TWAP, VWAP and Accelerator Oscillator strategy on 14 U.S equities, similar to Ning et al. (2018), they do not take into account the market impact correctly.

So far researchers have made a lot of progress in applying state-of-the-art techniques to the optimal execution-cost control problem, but they have neglected the very essential initial steps which are to evaluate representativeness of their dataset to the real data and to capture market impacts more realistically. Moreover, Q-learning was proved to converge to the optimal policy by Watkins and Dayan (1992) while deep Q-learning slowly converges if it does. While deep Q-learning is beneficial for problems with continuous state space, it typically requires a large amount of data to converge and its theoretical foundation is not well understood. For instance, for ATARI game, it's often necessary to wait for 10 to 40 million

[1] For further details of deep Q-learning, please refer to the very first introduction of deep Q-learning in Mnih et al. (2013)

frames to see significantly better results than random policy (Mnih et al., 2015). This nuisance gives rise to the dilemma of historical and simulated data tradeoff. Using a large volume of historical data risks the chance that market dynamics changes whereas simulated data may not represent the market dynamics well due to modeling error. Simulating limit order book is another challenging task to be addressed as discussed in detail below. Most of the studies employing deep Q-learning framework use simulated data to train the agent without discussing the goodness of fit of their simulated data compared with real data. Hence, in this study, I revisit the optimal trading execution problem using Q-learning, provide the essentials in Markov Decision Process formulation, introduce the presence of market impact of both market and limit orders into Q-learning and evaluate performance of the model based on historical data and simulated data. Deep Q-learning will be very much the next research topic.

2.3. Price Impact of Market and Limit Orders

Price impact has been extensively studied in the literature but there is little consensus on how to model it. In the empirical literature however, there seems to be an agreement that prices are moved by demand and supply imbalance. Most of the empirical studies have focused on the price impact of trades. Evans and Lyons (2002), Potters and Bouchaud (2003), and Kempf and Korn (1999) among others find that price impact of trades is an increasing and/or concave function of their size but they do not consider the driving force of quotes onto price formation. In the study of Huberman and Stanzl (2004), the authors conclude that arbitrage opportunities exist if the price impact of trades is permanent and non-linear. Gatheral (2010) extend the work of Huberman and Stanzl (2004) by showing that if the price impact of trades is non-linear, it needs to decay in a particular fashion to remove arbitrage. Bouchaud et al. (2010) associate the decay of market order price impact with incoming limit orders and argue that limit orders drive the prices towards the equilibrium, thereby offsetting the persistence in market order flow. While trades certainly play a crucial role in price movements, there is ample evidence that limit orders significantly affect the price. Arriving limit orders are found to reduce the price impact of trades and the reduction depends on the market depth (Weber and Rosenow, 2005; Hasbrouck and Seppi, 2001). Hopman (2007) emphasizes the driving force of supply and demand on stock prices and shows that the imbalance between buy and sell orders explains most of the stock price changes due to either uninformed price pressure or private information. Bouchaud (2009) shows that price impact of trades is neither linear in volume nor permanent as suggested by Kyle (1985) but rather strongly concave in volume and transient. Cont et al. (2014) reveals that on short time scale, order flow imbalance mainly drives price changes and there is a linear relation between order flow imbalance and price changes. They also argue that this linear relation implies the observed concave relation between price changes and trading volume albeit noisy and less robust than the relation of price changes with order flow imbalance. For these reasons, I will model the price impact of limit and market orders as the effect of changes in order flow imbalance and other important factors in order book on stock price.

2.4. Order Book Modeling – Simulation Approach

Order book modeling has been extensively researched over the last two decades due to the widespread use of algorithmic trading. The application of order book modeling is ample, especially in profit-cost analysis of complex trading strategies. There are two main approaches to order book modeling, one led by economists modeling the interactions between strategically rational agents and the other led by mathematicians assuming the agent acts randomly. The latter approach is often referred to as zero-intelligence since it assumes order arrivals and placements are stochastic. In the seminal paper of Smith et al. (2003), they study the properties of limit order book under the assumption of independent Poisson processes, then enriched with more realistic work such as Cont et al. (2010), Abergel and Jedidi (2013). However, Poisson process for the order arrival is not compatible with the actual data (Chakraborti et al., 2011) due to temporal dependencies between order arrivals. Recent studies by (e.g. Bacry and Muzy, 2014; Bacry et al., 2013; Bacry et al., 2016; Lu and Abergel, 2017) have proposed Hawkes process to order book modeling in an attempt to solve the dependency issue of order arrivals. Even though there have been tremendous efforts in investigating the problem of order book modeling currently, there are still some missing elements in the available models such as market participants' intelligence or explanation of empirical shape of order book. However, most of the empirical studies in optimal trading execution which employ order book modeling to simulate training data rely on basic techniques of order book modeling such as Zero intelligence or even simpler. This thesis is not to try to make contributions to this strand of research but rather illustrate the role and the drawbacks of using simulated data to train the agent with reinforcement learning techniques.

III. Datasets and Summary Statistics

This section describes the datasets used in this study, construction of main variables and summary statistics of key variables.

3.1. Datasets

My dataset consists of one calendar month (October 2019) of displayed order book data for 15 stocks traded on Nasdaq provided by Trading Physics. Trading Physics collects Nasdaq ITCH market data and sells it to users. These stocks are randomly selected based on average daily trading volume to examine how performance of policies found by Q-learning vary with liquidity levels. Due to the necessity to backtest, I equally divide the dataset into disjoint training and test sets. The agent will be trained based on the training set and the performance of the policies will be examined on the test set. It would be more realistic to train the agent and examine performance of the policies on a rolling fashion but due to time constraint, I just train the agent once on the training set and evaluate the policies once on the test set.

3.2. Variables of interest

3.2.1. Mid-price changes

Mid-price changes are changes in mid-price, calculated in number of ticks.

$$\Delta P_k = (P_k - P_{k-1}) / \delta$$

Where P_k is mid-price at time t_k and δ is tick size (in this dataset, equal to 1 cent)

Mid-price is the average of bid quote and ask quote at time t_k

$$P_k = \frac{P_{bid,k} + P_{ask,k}}{2}$$

3.2.2. Order Flow Imbalance (OFI)

Following Cont et al. (2014), I compute order flow imbalance based on Level I Price, which means limit orders sitting at the best quote. Level I Price is a reasonable choice since it directly affects price movement. Bid quotes represent demand for a stock while ask quotes represent supply for that stock. The demand-supply imbalance can be signaled by one of the following events:

- $P_{bid,n} > P_{bid,n-1}$ or $Q_{bid,n} > Q_{bid,n-1}$: a demand increase
- $P_{bid,n} < P_{bid,n-1}$ or $Q_{bid,n} < Q_{bid,n-1}$: a demand decrease
- $P_{ask,n} > P_{ask,n-1}$ or $Q_{ask,n} > Q_{ask,n-1}$: a supply increase
- $P_{ask,n} < P_{ask,n-1}$ or $Q_{ask,n} < Q_{ask,n-1}$: a supply decrease

Where $P_{bid,n}$ and $P_{ask,n}$ are the best bid and ask quote as of the n-th event. $Q_{bid,n}$ and $Q_{ask,n}$ are the size of the best bid and ask quote as of the n-th event.

Hence, the contribution of the n-th event to the demand-supply imbalance would be:

$$e_n = I_{(P_{bid,n} \geq P_{bid,n-1})} Q_{bid,n} - I_{(P_{bid,n} \leq P_{bid,n-1})} Q_{bid,n-1} - I_{(P_{ask,n} \geq P_{ask,n-1})} Q_{ask,n} + I_{(P_{ask,n} \leq P_{ask,n-1})} Q_{ask,n-1}$$

Notice that if the best bid quote stays the same, $\mathrm{e_n}$ reflects the increase in size added to the bid queue. If $\mathrm{e_n}$ is negative, part of the size at the best bid queue is canceled or executed with market sell orders. If the best bid quote rises as of the n-the event, $\mathrm{e_n}$ is equal to the size of price-improving limit bid order. On the other hand, if the best bid quote decreases, $\mathrm{e_n}$ equals minus the size of the best bid quote removed due to cancellation or market order. The same rule applies to the ask side with the reversed sign. The order flow imbalance over an interval $[t_{k-1}, t_k]$ is the sum of all contributions of all the events over this interval:

$$OFI_k = \sum_{n = N(t_{k-1})+1}^{N(t_k)} \mathrm{e_n}$$

Where $N(t_{k-1}) + 1$ is the index of the first event during the interval $[t_{k-1}, t_k]$ and $N(t_k)$ is the index of the last event during this interval.

3.2.3. Order Imbalance

Order imbalance is the imbalance between limit bid and ask orders for a security at the best quote. Material changes in order imbalance may reflect exogeneous shocks to the order placement due to news or external events. Order imbalance in a given period is sum of all order imbalance during that period whereas order flow imbalance is sum of all order imbalance changes after every event in that period. Thus, order flow depicts the overall net order flow while order flow imbalance tracks the changes in net order flow during every period. Order imbalance for an interval $[t_{k-1}, t_k]$ is computed as:

$$OI_k \ = \ \sum_{n\,=\,N(t_{k-1})+1}^{N(t_k)} \mathrm{Q}_{\mathrm{bid,n}} \ - \ \sum_{n\,=\,N(t_{k-1})+1}^{N(t_k)} \mathrm{Q}_{\mathrm{ask,n}}$$

3.2.4. Trade Imbalance

Unlike order flow imbalance, trade imbalance only considers trades but not limit orders and cancelations. Trade imbalance reflects the imbalance between total bid trade size and total ask trade size during an interval.

$$TI_k \ = \ \sum_{n\,=\,N(t_{k-1})+1}^{N(t_k)} \mathrm{Q}_{\mathrm{bid\ trade,n}} \ - \ \sum_{n\,=\,N(t_{k-1})+1}^{N(t_k)} \mathrm{Q}_{\mathrm{ask\ trade,n}}$$

3.2.5. Trading Volume

Trade volume on the other hand represents the overall activity of a security. Trade volume is a technical indicator signaling the current trend of the security's price. If trading volume is on the rise, the stock is increasingly demanded and the stock price is accordingly likely to move in the same direction for a short period of time. Trade volume is the sum of all bid trade size and ask trade size of all the trade events during a given interval.

$$TV_k \ = \ \sum_{n\,=\,N(t_{k-1})+1}^{N(t_k)} \mathrm{Q}_{\mathrm{bid\ trade,n}} \ + \ \sum_{n\,=\,N(t_{k-1})+1}^{N(t_k)} \mathrm{Q}_{\mathrm{ask\ trade,n}}$$

3.2.6. Average Market Depth

Market depth is often considered to reflect the market's ability to absorb market orders without significant market impact. In this study, average market depth refers to the average of total size of all limit orders at the best quote over the given interval $[t_{k-1}, t_k]$:

$$AD_k \ = \ \frac{1}{2(N(t_k) - N(t_{k-1}) - 1)} \sum_{n\,=\,N(t_{k-1})+1}^{N(t_k)} (\mathrm{Q}_{\mathrm{bid,n}} + \ \mathrm{Q}_{\mathrm{ask,n}})$$

3.2.7. Moving Difference

Moving average is the rolling mean of a time series over a specific bandwidth. Moving average is a well-known technical indicator that is often used to identify the short-term trend direction of a security. Moving difference is the difference between current mid-price and moving average of mid-prices.

$$MD_k \ = \ P_k \ - \frac{1}{n} \sum_{i\,=\,k-n}^{k-1} P_i$$

In this thesis, I choose the bandwidth n equal to 10.

3.2.8. Spread

Bid-ask spread is one measure of liquidity and of transaction cost. Spread is calculated as the difference between best ask quote and best bid quote.

$$Spread_k = P_{ask,k} - P_{bid,k}$$

3.2.9. Micro-price

Micro-price in the simplest form is weighted average price, expressed as an adjustment to the mid-price that considers bid-ask spread and order imbalance:

$$Micro - price_k = I_k P_{ask,k} + (1 - I_k) P_{bid,k}$$

Where weight I is provided with the best quote imbalance:

$$I_k = \frac{Q_{bid,k}}{Q_{bid,k} + Q_{ask,k}}$$

3.3. Summary Statistics

The table below presents summary statistics of main variables of interest grouped by every 30 seconds of the whole dataset for stock CATY. Noticeably, most of the variables are subject to high variance, especially OFI, spread, trading volume, trading imbalance and order imbalance.

	Count	Mean	Standard Deviation	25% percentile	50% percentile	75% percentile
Mid-price change	17941	0	0.035	-0.005	0	0.005
OFI	17941	3.444	829.176	-153	0	160
Average Market Depth	17941	130.382	97.940	82.423	124.765	169.182
Spread	17941	0.057	0.132	0.02	0.04	0.06
Trading Volume	17941	65.523	192.680	0	0	40
Trading Imbalance	17941	3.03	151.968	0	0	0
Order Imbalance	17941	-2.113	196.106	-99	0	99
Moving Difference	17941	0	0.036	-0.014	0.001	0.014

Table 1: Summary statistics of the whole dataset for stock CATY by 30 seconds

IV. Methodologies and Empirical Results

4.1. Price Impact

4.1.1. Model Specification

Following the work of Cont el al. (2014), I investigate the relation between price change, order flow imbalance and average market depth again and estimate price impact of limit orders and market orders as the expected price change due to the fluctuation of order books. Cont el al. (2014) propose the relation between price change, order flow imbalance and average market depth as:

$$\Delta\mathrm{P} = \frac{OFI}{2D} + \varepsilon$$

Where ΔP is price change; D is average market depth; ε is error term.

They argue that however, in reality, order books have more complications due to hidden orders not reported, intraday fluctuations of order book, or the influence of deeper levels. Hence, they propose the linear model specification of price impact.

$$\Delta P_k = \beta \, OFI_k + \varepsilon_k$$

Where β is the price impact coefficient.

In this study, I also examine the impact of other variables such as trading imbalance, order imbalance, average market depth and number of events happening over the interval and investigate non-linearities of trading imbalance (Kempf and Korn, 1999) and trading volume (Karpoff, 1987). The full model specification for price impacts in this study is:

$$\Delta P_k = \beta_1 \, OFI_k + \beta_2 \, TI_k + \beta_3 \, OI_k + \beta_4 \, MD_k + \beta_5 \, noEvent_k + \beta_6 \, TI_k \, |TI_k| + \beta_7 \, TV_k + \beta_8 \sqrt{TV_k} + \varepsilon_k \quad (1)$$

Then I pick the most relevant features among all of the features in this specification to make the model parsimonious for each stock and estimate the price impact due to order placements.

4.1.2. Data Processing

Before running the regression, I trim the outliers of variables of interest that are beyond the range of 95% confidence interval out of the dataset. The two images below show the dataset before and after being trimmed respectively. The number of observations decreases from 17941 to 11867 for the whole dataset. Data trimming is employed to reflect the average price impact coefficients without being influenced by statistical artifacts or influential observations. In reality for some market states, the price impact of consecutive placements or placement of a large volume may be much worse than the average price impact. Thus, the agent may penalize these actions less severely than it should under those circumstances. However, most importantly, the agent still incurs higher price impact due to trades than due to limit orders, due to large orders than due to small orders and due to consecutive placements than due to distant placements. This logic is essential to train the agent to select the optimal policy.

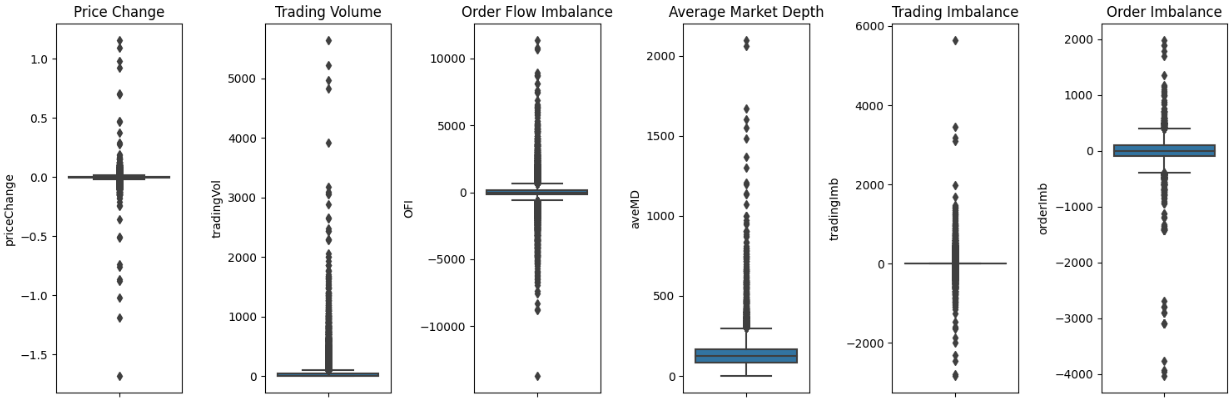

Figure 1: Variables of interest for price impact investigation before data trimming for stock CATY

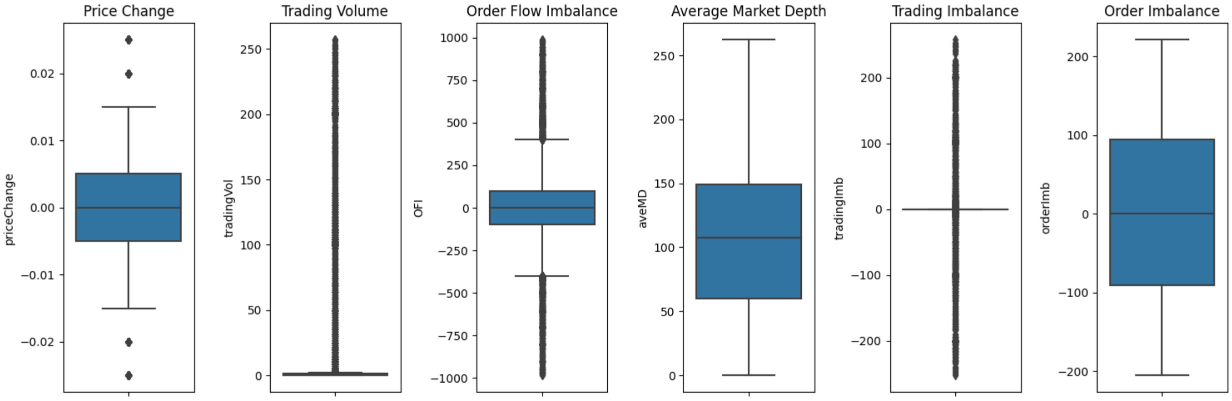

Figure 2: Variables of interest for price impact investigation after data trimming for stock CATY

Another important remark on understanding the dataset before empirical analysis of market impact is the correlation matrix of variables of interest. Understanding the correlation matrix is critical in the sense that it would prevent the problem of collinearity. Correlation matrix of variables of interest for price impact analysis is reported in table 2 in which TV, TI, OI and Ave MD denotes trading volume, trading imbalance, order imbalance and average market depth respectively. Observably, trading imbalance is highly correlated with squared trading imbalance and trading volume is highly correlated with square root of trading volume. The pairwise linear correlations of other variables are reasonable to avoid collinearity problem for regression.

	TV	OFI	Ave MD	TI	TI^2	$\sqrt{TV}$	OI
TV	1	0.01	0.3	0.13	0.21	0.87	-0.02
OFI	0.01	1	-0.02	0.21	0.15	0	0.16
Ave MD	0.3	-0.02	1	0.01	0.03	0.3	-0.15
TI	0.13	0.21	0.01	1	0.67	0.08	0.01
TI^2	0.21	0.15	0.03	0.67	1	0.09	0.02
$\sqrt{TV}$	0.87	0	0.3	0.08	0.09	1	-0.01
OI	-0.02	0.16	-0.15	0.01	0.02	-0.01	1

Table 2: Correlation matrix of variables of interest by 30 seconds on average for price impact analysis of stock CATY. TV denotes trading volume; OFI denotes order flow imbalance; ave MD denotes average market depth; TI denotes trading imbalance; OI denotes order imbalance.

4.1.3. Empirical Findings

The table below shows the results of different specifications from equation (1) for stock CATY. Model 1 to 3 are run to test the non-linearities of trading volume and trading imbalance found in other papers while model 4 and model 5 are a parsimonious version used for further analysis. Model 1 to 4 are analyzed based on the whole dataset while model 5 is analyzed based on only training set and is fed into Q-learning algorithm. The results of model 1 indicate the high statistical significance of order flow imbalance, trading imbalance and order imbalance, demonstrating the complementary power of these variables in explaining price changes. The findings from model 2 imply even though there exists the non-linear relation between price movement and trading imbalance, the linear relation is still stronger as indicated with higher t-statistics (28.121 > 23.334) and higher R-squared (24.6% > 21.8%). The results from model 3 essentially show no difference from those from model 1, meaning that neither trading volume nor squared root of trading volume is significantly associated with price movements. From all of the aforementioned findings, model 4 considers only variables of statistical significance, which are order flow imbalance, trading imbalance and order imbalance. The regression result from model 4 reveals an effectively parsimonious version of model 1 without any reduction in the goodness of fit. The statistical significance of order flow imbalance, trading imbalance and order imbalance are still substantially significant in model 5 with a dataset twice as small.

	(1)	(2)	(3)	(4)	(5)
Constant	-20***	-20***	-20***	-0.626	-0.289
	(-2.644)	(-2.495)	(-2.670)	(-0.854)	(-0.193)
OFI	1.102***	1.165***	1.102***	1.101***	0.950***
	(24.683)	(25.802)	(24.683)	(24.696)	(15.802)
Trading Volume	0.048	0.012			
	(0.247)	(0.605)			
Squared Root of			1.046		
Trading Volume			(0.438)		
Trading	5.872***		5.871***	5.875***	6.228***
Imbalance	(28.121)		(28.110)	(28.155)	(21.035)
Signed Trading		0.029***			
Imbalance		(23.334)			
Squared					
No. of events	-0.040	-0.016	-0.061		
	(-0.129)	(-0.050)	(-0.196)		
Order Imbalance	-0.618***	-0.630***	-0.619***	-0.617***	-0.655***
	(-8.852)	(-8.824)	(-8.852)	(-8.839)	(-4.854)
Average Market	0.157**	0.142**	0.152**		
Depth	(1.886)	(1.692)	(1.813)		
Adj. R-squared	24.6%	21.8%	24.6%	24.6%	22.8%
N	11.867	11.867	11.867	11.867	5003

Table 3: The regression results of different specification from equation (1) for stock CATY. Model 1 to 4 are analyzed based on the whole dataset while model 5 is analyzed based on the training set. All of the coefficients are scaled by 10^{-5}*. ***, ** and * indicate statistical significance at 99%, 95% and 90% confidence level respectively. T-statistics reported inside parenthesis is calculated based on heteroscedastic robust standard errors.*

To have a better grasp of the impact of trading activity on the relations between price movement and the variables of interest, I perform a similar empirical analysis over each half hour of trading day for stock CATY, shown in table 4. It should be noted that towards the end of a trading day, on average, the impact of three imbalance measures of order book, especially order flow imbalance, on price movement is increasingly significant. R-squared dramatically rises from 13.8% at the starting trading hour to 40.7% before the end of trading day. T-statistics of order flow imbalance also almost doubles and the magnitude of the coefficients of the three measures also drastically changes over the trading hours. Hence, it may be ideal to estimate the price changes due to the changes in order book over each half hour of trading day if there is a sufficient amount of data. However, due to the lack of data and prevention from overfitting, I will estimate the price impact based on the training set only for training, but based on the whole dataset for deriving the observed optimal policies. As the observed optimal policies are derived to define state space, it would be more robust to feed more recent data into the analysis.

4.2. Agent Training and Performance Evaluation

4.2.1. Markov Decision Process formulation of order execution

The optimal trading execution process can be formulated as a finite Markov Decision Process [2] with a terminal state. At the beginning of the episode, the agent is given an order to execute V shares in T minutes with either market or limit orders. The goal is to maximize the total expected returns from execution of V shares. For each of the subsequent time step since the arrival time, the agent observes the current state including its own private state and market state and acts on the current status. The agent will place a child order of a weakly smaller size than the size of its own remaining inventory with either market or limit order. The decision on volume and order type for the placement depends on the trade-off between trading urgency, immediate cash flow from the placement and price impact of the placement on subsequent placements. The assumption in this formulation is the Markov property which implies that the effect of an action at any given state is dependent on that state and not on the prior history.

[2] I suppose the readers already know Markov Decision Process and dynamic programming concept which are also discussed in optimal trading execution literature (e.g. Bertsimas and Lo, 1998)

	Constant	OFI	Trading Imbalance	Order Imbalance	Adj. R-squared	N
9:30 – 10:00	-10	2.462***	4.245***	-0.777***	13.8%	1120
	(-0.195)	(7.824)	(3.968)	(-3.803)		
10:00 – 10:30	9.446	1.025***	8.237***	-0.447	21.8%	1121
	(0.220)	(6.263)	(8.992)	(-1.133)		
10:30 – 11:00	-20	0.648***	4.426***	-0.589**	13.6%	1119
	(-0.524)	(4.379)	(5.004)	(-1.742)		
11:00 – 11:30	40	0.529***	4.587***	-0.904***	18.2%	1127
	(1.115)	(4.198)	(8.049)	(-3.404)		
11:30 – 12:00	-20	1.009***	3.9***	-0.558***	23.8%	1117
	(-0.765)	(7.507)	(6.703)	(-3.260)		
12:00 – 12:30	10	1.027***	3.484***	-0.676***	22.7%	1125
	(0.655)	(6.599)	(6.681)	(-3.965)		
12:30 – 13:00	-30*	1.352***	2.072***	-0.783***	24.4%	1128
	(-1.404)	(9.560)	(2.255)	(-5.562)		
13:00 – 13:30	-10	1.271***	1.666***	-0.258***	21.5%	1117
	(-0.883)	(10.241)	(3.889)	(-2.845)		
13:30 – 14:00	20	1.028***	1.99***	-0.285***	19.7%	1120
	(1.064)	(7.539)	(2.287)	(-2.870)		
14:00 – 14:30	-10	1.391***	2.266***	-0.540***	28.6%	1137
	(-0.864)	(9.461)	(3.675)	(-5.072)		
14:30 – 15:00	-6.78	1.385***	1.948***	-0.181***	32.7%	1123
	(-0.397)	(12.312)	(3.498)	(-2.639)		
15:00 – 15:30	40***	1.204***	1.552***	-0.416***	39.4%	1132
	(2.334)	(15.663)	(5.680)	(-4.434)		
15:30 – 16:00	-30**	0.988***	0.618***	-0.476***	40.7%	1121
	(1.699)	(14.520)	(3.088)	(-5.439)		

Table 4: The regression results of parsimonious model of price impact for stock CATY every half hour during a trading day in the whole dataset. All of the coefficients are scaled by 10^{-5}. ****, ** and * indicate statistical significance at 99%, 95% and 90% confidence level respectively. T-statistics reported inside parenthesis is calculated based on robust standard errors.*

The performance of the agent is benchmarked against the gain from execution of the standard strategy such as Submit and Leave, and TWAP. Submit and Leave strategy means placing a limit sell order at a fixed price, and executing the remaining unfilled shares with market order at the end of the episode. This strategy is one of the most simple strategies performed by traders and captures the reality of monitoring cost incurred by them when dealing with well diversified portfolios. TWAP is also a popular benchmark used by traders to evaluate their trading strategies due to their simplicity and effectiveness. TWAP stands for time-weighted-average-price, being a strategy of executing trades evenly in a given time period and is optimal under the assumption that price is a Brownian motion, and more generally a martingale (Bertsimas and Lo, 1998). I now give the MDP formulation employed in this study, describing the state, action, reward as well as the Q-learning algorithm.

States

In MDP, state space is the representation of the world, implying that changes in state space reflect all the critical changes in the real-world environment that trigger behavior of the agent. For simplicity, I consider only observed states, but not hidden states and treat a partially observable environment as if it were fully observable. State space includes private state and market state, where private state is exclusive to the agent while market state is common to all of the agents. Private state consists of two important variables, namely the remaining time and the remaining inventory representing how much time for the agent is left and how many remaining shares the agent is required to execute. To render the state space discrete, I equally split V shares into n lots, each containing V/n shares. The agent will submit a revised order of x number of lots every micro-interval with x smaller than n, thereby the remaining inventory ranging from 0 to n. Similarly, T time is also discretized into m micro-intervals.

Selection of market variables is less trivial and more challenging due to the hidden states and strategic interactions of market participants. In this study, I neglect subtle interactions of multiple agents or traders in the market but rather focus on the changes in order book that drive different behaviors of the agent. In an attempt to determine the market variables, I perform an empirical analysis exploring which variables in the order book drive the observed optimal policy of the agent. The observed optimal policy is specified to maximize the expected rewards given that the whole price trajectory is observed. I explore a number of market state presentations and select the ones with highest statistical significance. After the selection procedure, I discretize these market variables into a number of bins and validate their significance again. For simplicity, I employ the linear probability model for the empirical analysis which can be extended with non-linear model such as multinomial logit, least squared regression on non-linear features or deep learning. The critical assumption in this research is that the actions of the agent do not change the market state trajectory but only create the price impact on the immediate next quote. This assumption is more likely to fail to hold if the placement volume is large and reveal trading intentions of the agent.

Actions

As aforementioned, policies map states to actions and for Q-learning, action space also must be discrete. Action space in this research includes limit orders and market orders of a variety of volumes. In each of micro-interval, the agent can place a limit order or execute x number of lots but not simultaneously. Limit order placement and trade execution can be performed concurrently but would worsen the price impact and reveal trading intentions even more, thereby inclusion of this scenario appearing trivial. The agent places limit orders at the best quote since allowing the agent to place limit orders at other price levels would require a larger state space and demand exponentially more data. In each of the micro-interval, if the agent places a limit order and only a partial number of shares are filled, the agent will cancel the remaining number of shares in that limit order and select the next optimal action in next period with the remaining shares to be executed.

Rewards

There are two sources of rewards, namely immediate rewards and temporary price impact on the subsequent placement. Immediate rewards essentially are the proceeds from (partial) execution of orders. The shares filled are estimated as the opposite trade volume minus the outstanding limit order, rounded to a lot volume. For instance, at the beginning of time step 2, the agent places an ask limit order of 100 at price 35, the bid trade volume is 148 and the outstanding ask limit order at the best quote (35) is 100. Since each lot contains 50 shares, only 50 shares of the agent’s order are filled. While limit orders are not always filled, market orders always are, for simplicity at the best opposite quote. The setting can easily be extended to the more realistic scenario such as all shares in the market orders may not be executed at the best quotes, but similarly to extension of limit order placement at other price levels, allowing market orders to be filled at other price levels also requires a larger state space and thereby exponentially more data to achieve reasonable estimates. The agent will reach the absorbing terminal state either when he executes all the shares within T minutes or at the end of T minutes. In case there is remaining inventory at the end of T minutes, all of these remaining shares will be executed with market orders at the end of T minutes.

To allow comparisons of policies across time and across stocks, which may be subject to fixed time effects and fixed stock effects, the execution price to compute immediate rewards are benchmarked against the mid-price at the beginning of each episode. Hence, the performance of the learned policy is effectively compared with that of the ideal policy which involves execution of V shares at the mid-quote, assuming infinite liquidity at the arrival time. Since this ideal policy cannot be realized, the learned policy is always expected to underperform this ideal policy. Therefore, the immediate rewards can be regarded as an underperformance measure of the learned optimal policy compared to the mid-price baseline policy.

Price impact is described in detail in the earlier section. It is worth emphasizing that price impact is only temporary, thereby affecting only the immediate subsequent quote. As shown earlier, the price impact of trades is essentially more significant than that of limit orders, but price impact of a small trade may not be larger than that of a sizable limit order due to information revelation or huge misbalance on the limit order books. Since my study is primarily targeted to institutional passive investors, exchange fees and commission can be assumed negligible and price impacts are often found temporary. For simplicity, I also do not take the impacts of fee schedule of stock exchanges into consideration and this study accordingly applies to submission of orders to stock exchange without fee schedule. Due to the lack of data, I do not consider latency which involves the arrival delay of orders to the exchanges.

4.2.2. Learning Algorithm – Q-learning

Q-learning is off-policy temporal-difference control learning. Temporal-difference (TD) learning is one of the most central and novel ideas proposed to reinforcement learning.

TD learning is a mix of Monte Carlo and dynamic programing in the sense that similar to Monte Carlo methods, it learns directly from original experience without a model of dynamics of the environment while it bootstraps based on the previously learned estimates just like dynamic programming.

4.2.2.1. TD learning and its advantages

In a finite-horizon MDP setting, the objective of learning is to maximize the expected cumulative rewards the agent receives in the long run. If rewards at time step t is denoted as R_t and γ is the discount rate ($0 \leq \gamma \leq 1$), the expected cumulative return for a T-minute episode is:

$$G_t \; = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \; = \; R_{t+1} \; + \; \gamma R_{t+2} \; + \; \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-t} R_T \; = \; R_{t+1} \; + \; \gamma G_{t+1}$$

The episodic task requires a series of episodes, each of which comprises a finite sequence of time steps. In reinforcement learning, state-value function of state s under policy π, denoted as $V_\pi(s)$ is the expected return starting at state s and following the policy π thereafter:

$$\begin{aligned} V_\pi(s) \; &= \; E_\pi(G_t \mid S_t = s) \qquad (2) \\ &= \; E_\pi(R_t + \; \gamma G_{t+1} \mid S_t = s) \\ &= \; E_\pi(R_t + \; \gamma V_\pi(S_{t+1}) \mid S_t = s) \quad (3) \end{aligned}$$

State-action value function or Q-function is defined as expected return gained from starting at state s, taking an action a and following the policy π thereafter:

$$Q_\pi(s, a) \; = \; E_\pi(G_t \mid S_t = s, A_t = a)$$

State-value and state-action value can be estimated from historical data. If each state is visited infinitely, by the law of large number, on average these two functions will converge. The averaging procedure over many random samples of actual returns is called Monte Carlo method. A simple every-visit Monte Carlo [3]algorithm can be written as:

$$\mathrm{V}(S_t) \; \leftarrow \; \mathrm{V}(S_t) \; + \; \alpha \, [G_t \; - \; \mathrm{V}(S_t)]$$

[3] Every-visit Monte Carlo method updates the state value for every visit of state S. For further details of Monte Carlo method, please refer to Sutton and Barto (2018).

Where α is a constant step size parameter or a fixed learning rate, G_t is total expected return starting at time t and $\mathrm{V}(S_t)$ is state-value function of state S at time t. I will discuss the necessity of a fixed learning rate under the nonstationary environment in the following section. Whereas Monte Carlo method needs to wait until the end of the episode to update (since G_t must be known), TD method bootstraps with only the value function of the next step. A simple one-step TD method updates:

$$\mathrm{V}(S_t) \leftarrow \mathrm{V}(S_t) + \alpha\,[R_t + \gamma \mathrm{V}(S_{t+1}) - \mathrm{V}(S_t)]$$

Monte-Carlo methods use an estimate of equation (2) as a target while dynamic programming methods target the estimate of equation (3). Monte-Carlo target is an estimate since the expected return is unknown and estimated with the sample return. Dynamic programming target, on the other hand, is an estimate since the expected value of $V_{\pi}(S_{t+1})$ is replaced with its current estimate. The expected value of the whole term in equation (3) can be calculated based on the assumption of dynamic programming about a model of the environment. TD target is an estimate for the same reasons that similar to dynamic programming, it replaces the true $V_{\pi}(S_{t+1})$ with its current estimate and similar to Monte-Carlo, it samples the expected value in equation (3).

Combining Monte-Carlo with dynamic programming methods, TD methods have advantages over these two. Noticeably, TD methods are more advantageous than dynamic programming in that they do not require a perfect model of the environment, reward and transition probability distribution. As compared to Monte-Carlo, TD methods are more beneficial in that they update in an incremental fashion which is critical to cases of long episode or experimental actions. Both TD and Monte-Carlo methods are proved to asymptotically converge in the mean for a small constant step size, but TD methods are biased due to bootstrapping whereas Monte-Carlo methods are not (Sutton and Barto, 2018).

4.2.2.2. Learning Rate and Convergence Property

The averaging methods are appropriate under the stationary environment, that is the reward probabilities do not change over time. However, time series is often nonstationary and therefore more recent rewards may be more relevant than long-past ones. One of the tricks to deal with the nonstationary environment is to use a fixed learning rate. The fixed learning rate must be sufficiently small for value function of TD methods to converge in the mean (Sutton and Barto, 2018). In order for the value function to converge with probability 1, the step size must satisfy the following stochastic approximation conditions (Watkins and Dayan, 1992):

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty \quad \text{and} \quad \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty$$

The first condition implies that the step sizes are large enough for the algorithm to deal with the initial fluctuations. The second condition is required to guarantee that the step sizes are small enough to ascertain convergence. In case of sample average, $\alpha_n(a) = \frac{1}{n}$, these two conditions are met but if $\alpha_n(a)$ is a constant, the estimates of value function will never completely converge but vary in response to the most recent rewards, which is desirable in nonstationary environment.

4.2.2.3. Q-learning

Q-learning is an off-policy TD control algorithm, introduced by Watkins (1989) and proved by Watkins and Dayan (1992) to converge under the condition that all state-action pairs are updated infinitely and under the aforementioned conditions of learning rate as for TD methods in general. Instead of using state-value function as above, Q-learning updates state-action value estimates and lets the agent to select the next optimal action based on the current estimate. Standard Q-learning algorithm makes an update as:

$$\mathrm{Q}(S,A) \leftarrow \mathrm{Q}(S,A) + \alpha\,[R + \gamma max_a \mathrm{Q}(S',a) - \mathrm{Q}(S,A)]$$

The optimal policy is searched to maximize the total expected rewards:

$$\pi^* = argmax_\pi Q^\pi(s,a)$$

It should be noted that even during a short time interval, if the investor wants to incorporate trading urgency into the algorithm, he could set discount rate γ positive or apply any utility function with parameter γ corresponding with his risk appetite since discount rate naturally reflects the time value of money under the investor’s perspective. The lower the discount rate is, the higher utility the immediate rewards bring. The discount rate should be proportionally inverse to the stock volatility, depending on the investor’s risk aversion.

In an attempt to include the price impact on the immediate next quote, I adjust the standard Q-learning as:

$$\mathrm{Q}(S,A) \leftarrow \mathrm{Q}(S,A) + \alpha\,[R + \gamma max_a(\mathrm{Q}(S',a) + PI_A\, a) - \mathrm{Q}(S,A)]$$

Where PI_A is the price impact caused by action A. The idea is that if the agent takes an action A, it will create a price impact on the subsequent quote and the total price impact will be amplified by the immediate subsequent action. It is optimal for the agent to select the best next action based on the total price impact and current estimate of Q-value of the next time state.

Below is the pseudo-code of the learning algorithm:

```
Initialization: $\forall\ visited\ s \in S, a \in A, Q(s,a) = 0$
For remaining time t from 0 to T:
        For remaining inventory i from 0 to V:
        Set state = {t, i, $o_1$, $o_2$, …, $o_R$}
                For action a from 0 to i:
                        Simulate state s to state $s'$
                        Calculate price impact for action a
                        Update:
                        $\mathrm{Q}(s,a) \leftarrow \mathrm{Q}(s,a) + \alpha\,[R + \gamma max_{a'}(\mathrm{Q}(S',a') + PI_A\, a') - \mathrm{Q}(s,a)]$
Optimal policy for every state: $\pi^*(s) = argmax_a Q^*(s,a)$
```

4.2.3. Q-learning with Experience Replay

In the case of continuous state space, tabular Q-learning cannot be used and replaced with function approximation to estimate state-action Q values, two popular methods of which are linear approximation and Deep Q-learning. Q-learning also can be learned with online policy which involves updating the estimates continuously with a new stream of experience and discarding all of the previous data. This update form has two issues: (1) the independent and identical distribution assumption of many popular stochastic gradient-based algorithms used in function approximation is broken due to strongly correlated updates, and (2) rare experience which may be useful later is quickly forgotten. Lin (1992) introduces experience replay to address these two issues. He proposes to store experience in a replay memory and mix more recent with less recent experience for the updates. This operation helps break the autocorrelation of observations across time and reduce the required amount of experience for learning. Mihn et al. (2013, 2015) use uniformly random experience replay within their Deep Q-Network. Schaul et al. (2015) continue the work and propose prioritized experience replay which involves replaying the transitions with high expected learning progress measured by TD error more frequently. This prioritization process may lead to a loss of diversity and introduce bias which may be alleviated with stochastic prioritization and importance sampling (Schaul et al., 2015). In particular, the agent stores the most recent K experience samples of current state, action, reward, next state as transition tuples and update the probability of sampling transition i as:

$$\mathrm{P(i)} = \frac{p_i^\alpha}{\sum_{k=1}^{K} p_k^\alpha}$$

Where $p_i > 0$ is the priority of transition i or in other words, the absolute value of TD error of transition i and α is the prioritization level. $\alpha = 0$ corresponds to the uniformly random sampling.

Experience replay can also be applied to off-policy tabular Q-learning with a little modification to use data more efficiently. Instead of sampling from most recent observations, I sample from most recent K episodes. The TD error of each episode is the maximum of TD errors of all the transitions during that episode. The idea is that the whole episode is more likely to be updated if a transition in this episode has higher priority over the other transitions in other episodes. However, it is difficult to incorporate importance sampling weight[4] into tabular framework to anneal the bias of prioritized replay. Thus, I do not include the importance sampling weight in the algorithm below and let the agent sample only one episode for every episode due to the time constraint of this study.

[4] For further details of how importance sampling weight embedded into Q-learning with function approximation, please refer to (Schaul et al., 2015).

```
Initialization: $\forall\ visited\ s \in S, a \in A, Q(s, a) = 0$, set K and $\alpha$
For each episode e in training set:
    Calculate TD error for each transition in the episode:
        $TD_i = R + \gamma max_{a'}(\mathrm{Q}(S', a') + PI_A\ a') - \mathrm{Q}(s, a)$
    Calculate priority of the episode:
        $p_e = max_i|TD_i|$
    Update the probability of sampling K episodes:
        $\mathrm{P(e)} = \frac{p_e^{\alpha}}{\sum_{k=1}^{K} p_k^{\alpha}}$
    Sample one episode from these relative probabilities and update the episode:
Optimal policy for every state: $\pi^*(s) = argmax_a Q^*(s, a)$
```

4.2.4. Specification of State Space

For the following analyses, without loss of generality, the agent is given an order to sell 500 shares in 5 minutes, which is a set of 30-second micro-intervals and the agent is risk-neutral, implying that the discount rate equals 1. One of the justifications for letting the agent act every 30 seconds is the vanishing autocorrelations of price change and order flow imbalances after 30 seconds on average. In this setting, the problem is much simpler in the sense that it is approximately sufficient to only account for temporary price impact of the agent's placement on average. It may not be entirely realistic since the large order flow imbalance may affect the subsequent order flow imbalance but the minor order flow imbalance may not. Thus, the algorithm used in this study may penalize the order that significantly changes the order flow imbalance less than it should and penalize the order that do not significantly affect the order flow imbalance more than it should, but data trimming before estimation of price impact alleviate this bias to a considerable extent. Investors can choose different micro-interval depending on their target but should be reminded that the more frequently the agent acts, the more time steps the market impact echoes over.

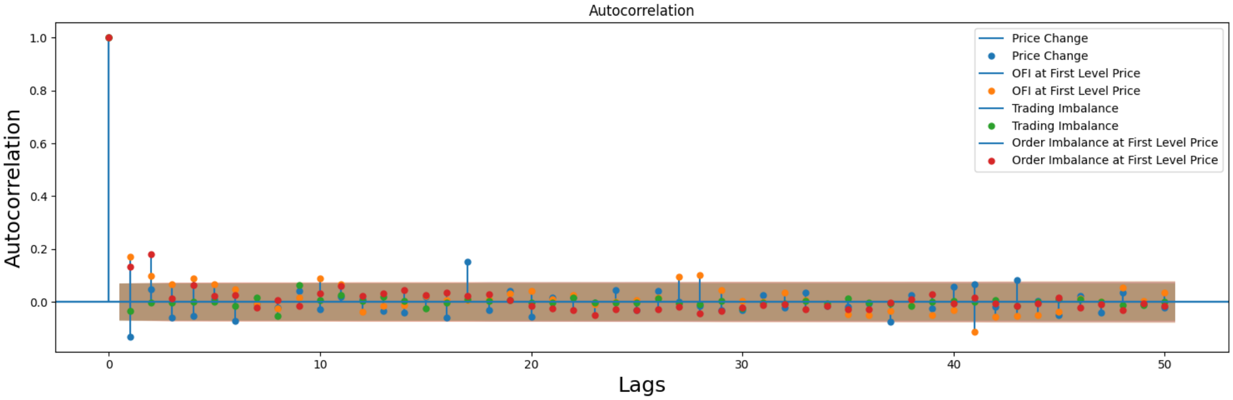

Figure 3: Autocorrelation graph for price change, order flow imbalance, trading imbalance and order imbalance every 30 seconds on average for stock CATY.

As mentioned earlier, state space is determined based on the observed optimal policy on the whole dataset. I perform an empirical analysis exploring which variables drive the observed optimal policy of the agent using linear probability model. The observed optimal policies are defined in detail in the below section.

4.2.4.1. Observed Optimal Policies

The observed optimal policies are derived based on the total expected rewards under the assumption that the price trajectory over T-minute periods is observed. I illustrate the process of deriving optimal policies for execution of 500 shares in 5 minutes as in table 5. Again, as each time step lasts 30 seconds, a 5-minute episode comprises of 10 time steps. Assuming that the agent can observe the price trajectory of the whole episode and he is risk-neutral (or he has no urgent need of trading), he would choose to place a sell limit order of 200 shares at the best ask quote (38.93) at the first time step and execute a sell trade of 300 shares at the best bid quote (36.27) at time step 8 to maximize his accumulated return. It should be noted that since the agent is risk-neutral, over a very short time horizon, the discount rate is essentially 1, maximization of total expected return is equivalent to maximization of accumulated return. In this 5-minute episode, observably the agent gains the most out of a share from placing a limit order at the first time step if it is filled since the best ask quote at the first time step is 38.93, higher than any of the quote during this episode. The agent will incur a price impact for the next immediate order if he chooses to act in the next step, but he chooses not to. Thus, there is no price impact for his placement at time step 1.

Time Step	Best Bid Price	Best Ask Price	Ask Vol Filled	Const	OFI Coeff	TI Coeff	OI Coeff	Optimal Action Type	Optimal Volume
1	36.03	38.93	200	-0.124	0.025	0.042	-0.008	Limit	200
2	35.60	36.00	0	-0.124	0.025	0.042	-0.008	None	0
3	35.65	36.00	0	-0.124	0.025	0.042	-0.008	None	0
4	35.75	36.00	0	-0.124	0.025	0.042	-0.008	None	0
5	35.75	36.00	1550	-0.124	0.025	0.042	-0.008	None	0
6	36.27	36.43	0	-0.124	0.025	0.042	-0.008	None	0
7	36.27	36.44	0	-0.124	0.025	0.042	-0.008	None	0
8	36.27	36.43	0	-0.124	0.025	0.042	-0.008	Market	300
9	36.18	36.23	0	-0.124	0.025	0.042	-0.008	None	0
10	36.12	36.21	0	-0.124	0.025	0.042	-0.008	None	0

Table 5: An illustration of observed optimal policy in one episode. OFI denotes order flow imbalance. TI denotes trading imbalance. OI denotes order flow imbalance. Const, OFI coefficient, trading imbalance coefficient and order imbalance coefficient are scaled by 1/1000.

From time step 2 to time step 10, he executes a sell trade of 300 shares at price 36.27 since it is the highest best bid quote over the remaining time steps and even higher than the best ask quote at time step 5. The reason for the agent to act in step 8 instead of step 6 is that the distance between his first action at step 1 and his next action at step 8 is longer than that between step 1 and step 6, thereby the price impact due to his first action more likely to disappear at step 8 than at step 6. At this point, he reaches the absorbing terminal state as he manages to execute all of the shares in the given order. If he chose to place an order at time step 2, he would incur the price impact at this time step, which is equal to:

$$\text{Const} + \text{OFI Coefficient} \times \text{OFI Change} + \text{TI Coefficient} \times \text{TI Change}$$

$$+ \text{OI Coefficient} \times \text{OI Change} \quad (4)$$

Where OFI Change, TI Change and OI Change are change in OFI, trading imbalance and order imbalance respectively due to the agent’s order at time step 1.

4.2.4.2. Specification of State Space

To specify state space, I run the regression of all actions in action space including market orders and limit orders with a variety of volume levels (model 1), trades with different volumes (model 2), limit orders with various placement volumes (model 3) and types of orders (model 4) on many variables of order books and time step. It is worth noting that the reason for inclusion of the best bid volume into the analysis is that the agent is attempting to sell shares and the best bid volume therefore is the volume of the opposite best quote, which may affect the fill probability of limit order in the subsequent step. Model 1 considers all of the actions including the action when the agent stays put whereas the remaining three models only take into account the actions with positive volume. Thus, the model 1 examines the relation of all the variables in the order book and the observed optimal behavior of the agent while the model 2 and 3 investigate which variables drive the placement volume of the agent if he decides to execute a trade or place a limit order respectively. The model 4, on the other hand, aims at finding the relation of the agent's decision on order types and the variables in the order book if he opts to act. In an attempt to exclude influential observations that are not very representative of the data and may distort the empirical results, I trim the most heavily tailed variables, namely OFI, trading volume and spread, beyond the 95% confidence interval out of the dataset to perform the first empirical analysis in this section but not the discretized version since discretization renders trimming unnecessary. Additionally, as all of the independent variables have huge differences in magnitude, I scale each variable by the formulation:

$$X_i' = \frac{X_i - min(\boldsymbol{X})}{max(\boldsymbol{X}) - min(\boldsymbol{X})}$$

The results in model 1 show that OFI, average market depth, spread, micro-price, order imbalance, time step, and trading volume have statistically significant impacts on the decision of the agent on which action to take. The results in model 2 imply only trading imbalance, moving difference and spread significantly influence the volume choice of the agent if it chooses to execute trades. In case the agent places a limit order, spread, micro-price, time step, trading volume, and moving imbalance have a statistically significant influence on how the agent chooses placement volume, as indicated by the results of model 3. Model 4 shows OFI, spread, order imbalance, time step, and moving difference significantly affect the agent’s decision on types of order. If there is enough data with rich variation in state space, it would be ideal to determine state space with the training set only to avoid the overfitting problem. Even though with a humble amount of data, I find similar results of the analysis based on the training set but I report the results based on the whole dataset in the table 6.

Then I discretize the most statistically significant continuous market variables, namely spread, trading volume, trading imbalance, order imbalance, and moving difference, into three quantile bins, except for trading volume and trading imbalance which are mostly 0 and require additional care. Noticeably, micro-price and OFI have a significant impact on the agent’s overall action profile but trading imbalance only affects the agent’s decision on trade volume. The reason for inclusion of trading imbalance instead of micro-price or OFI is due to the fact that just a few of the factors considered have a significant influence on how the agent changes the trade volume, among of which is trading imbalance while other factors contribute significantly to the explanation of how the agent chooses limit order volume or order type. Due to the computational expense and lack of data, I only select the most statistically significant variables to explain different behavior of the agent. It is noteworthy that spread, order imbalance, and moving difference are discretized into 3 bins, namely from negative infinity to 30% quantile, 30% quantile to 70% quantile and 70% quantile to positive infinity while trading volume and trading imbalance are divided into 3 bins, namely from negative infinity to 0, 0 to 100, 100 to positive infinity. To evaluate the discretization process, I run the same regression with these discretized variables instead as shown in table 7. Notably, with discretization and more parsimonious models, R-squared of all models and statistical significance of variables are all improved. This implies discretization well represents the changes in variables that drive different optimal behavior of the agent.

Non-linear models can be performed to explore the non-linear relation between the observed optimal policy and variables of order book. However, with linear probability model, I manage to determine some of the crucial variables that drive different optimal behavior of the agent. Including more variables into the Q-learning would be more computationally expensive and require a significantly more amount of data for the estimates of the state-action values to converge. For these reasons, linear probability model may be sufficient in this study.

Dependent variables	All Actions (1)	Market Orders (2)	Limit Orders (3)	Type of Orders (4)
Const	9.0431*** (20.965)	9.0827*** (13.368)	13.5315*** (9.417)	-0.2711 (-0.625)
OFI	-0.2839*** (-2.233)	0.0903 (0.351)	0.4044 (0.546)	-0.1643*** (-2.494)
Average MD	-0.6801** (-1.779)	-0.3636 (-0.766)	-1.5970* (-1.293)	-0.1120 (-0.695)
Spread	0.6367*** (7.204)	-0.5459*** (-2.697)	5.3381*** (6.508)	0.4200*** (6.847)
Micro-price	0.1246** (1.902)	0.1022 (0.971)	0.9142** (2.305)	0.0307 (0.956)
Order Imbalance	3.4140*** (6.243)	0.7685 (0.814)	1.6159 (0.700)	2.0523*** (3.453)
Time Step	-0.7262*** (-13.483)	-0.0522 (-0.560)	-1.9883*** (-5.495)	-0.3561*** (-14.304)
Trading Volume	-0.1389** (-1.748)	-0.2056 (-1.061)	-1.1659** (-1.767)	0.0455 (0.895)
Trading Imbalance	0.1336 (0.807)	-0.7888** (-2.070)	-0.6067 (-0.464)	0.0608 (0.594)
Moving Diff	0.2773 (0.808)	0.9219*** (2.400)	7.1613*** (5.725)	-1.1424*** (-9.186)
Best Bid Volume	-0.4618* (-1.268)	-0.1225 (-0.242)	-3.0881* (-1.591)	0.0005 (0.002)
Adj. R-squared	1.9%	0.8%	12.5%	17.5%
N	14617	1499	901	2400

*Table 6: Regression results of observed optimal policies on order book variables and time step for stock CATY. ***, ** and * indicate statistical significance at 99%, 95% and 90% confidence level respectively. T-statistics reported inside parenthesis is calculated based on robust standard errors.*

Dependent variables	All Actions	Market Orders	Limit Orders	Types of Orders
Const	11.3159*** (279.161)	9.6067*** (119.814)	17.6643*** (53.982)	0.4378*** (18.309)
Discretized Trading Vol	-0.0163 (-0.714)	-0.0235 (-0.416)	-0.4017** (-2.159)	0.0312* (1.883)
Discretized Moving Diff	0.0652*** (2.602)	0.1063** (2.261)	0.9140*** (7.522)	-0.0996*** (-8.373)
Discretized Spread	0.2260*** (11.920)	-0.1416*** (-3.389)	1.3005*** (8.888)	0.1383*** (13.244)
Discretized Order Imb	0.1201*** (5.999)	0.0503* (1.349)	-0.2187** (-1.713)	0.0878*** (8.384)
Discretized Trading Imb	-0.0190 (-0.567)	-0.2551*** (-3.458)	-0.1595 (-0.699)	-0.0169 (-0.886)
Time Step	-0.0807*** (-14.962)	-0.0004 (-0.042)	-0.2233*** (-5.643)	-0.0380*** (-14.899)
Adj. R-squared	2.1%	2.3%	14.9%	17.8%
N	17940	1770	1108	2878

*Table 7: Regression results of observed optimal policies on discretized order book variables and time step in a parsimonious version for stock CATY. ***, ** and * indicate statistical significance at 99%, 95% and 90% confidence level respectively. T-statistics reported inside parenthesis is calculated based on robust standard errors.*

4.2.5. Benchmarks

As mentioned earlier, I employ two benchmarks such as Submit and Leave strategy and TWAP strategy to compare their performance with the performance of the strategies learned from Q-learning model. Submit and Leave strategy essentially involves placing a limit order at the first time step, waiting and submitting the remaining unfilled shares with market orders at the end of the episode. The agent incurs no price impact when implementing this strategy since the time from the first order to the second order is the whole episode, creating no temporary price impact. TWAP strategy is to execute trades of an equal volume during the whole episode. At each time step, following this strategy, the agent executes V/n shares where V is the number of total shares to be executed and n is the number of time steps in one episode. The agent incurs price impacts every time step from time step 2 to time step n due to execution of trades from time step 1 to time step n-1 respectively. The price impact is estimated as in equation 4.

4.2.6. Performance Evaluation Based on Historical Data

Normally in reality, learning rate should be chosen based on cross validation with validation dataset, but due to the lack of the data, I skip the cross-validation part and report the out-of-sample results of Q-learning policies with different learning rates and with experience replay in the table below. Column 2 of the table 8 indicates the average accumulated rewards of policies in test set. Column 3 presents the variance of policies compared to the observed optimal policy. Column 4 reports the modified Sharpe ratios of each strategy. Row 2 shows the performance of the optimal policy if the whole price trajectory is observed. Row 3 and 4 present the performance of the optimal policy learned from Q-learning with learning rate 0.1 and 0.3 respectively. As the training set consists of only two weeks of data, there may not enough data for the estimates to converge. Thus, I employ experience replay to make use of the available data more efficiently. I also examine the importance of temporal correlation by running the Q-learning algorithm with deterministic experience replay and random experience replay. The same experience is replayed with learning rate 0.1 for the both rounds as shown in row 5, and 0.3 for the first round and 0.1 for the second round as indicated in row 6. Row 7 and 8 report the performance of policies learned from Q-learning with random experience replay and replay buffer K equal to 100 episodes, row 7 with prioritized replay and row 8 with uniform replay. I also present the performance of Submit and Leave strategy and TWAP strategy in row 9 and 10 to compare the results of Q-learning policies with the benchmark policies. With the aim of comparing risk-return of all policies, I modify the Sharpe ratio to adjust for negative rewards as follows:

$$\text{Mod. Sharpe Ratio} = \frac{X + Average\ Rewards}{\sqrt{Variance}}$$

Sharpe ratio is meaningful only when the return of the strategy to be compared is higher than the risk-free rate. I illustrate the failure of standard Sharpe ratio with an instance: the first strategy has an average return of -0.1 and a variance of 4; the second strategy has the same average return but has a variance of 8; the third strategy has an average return of -0.2 and a variance of 8; the risk-free rate is 0.5. Sharpe ratio of the first strategy is -0.3, lower than that of the second strategy, -0.21 while the two strategies has the same average return but the first strategy has lower variance and therefore should be preferred to the second strategy. The third strategy has Sharpe ratio of -0.24, higher than that of the first strategy and lower than that of the second strategy but actually has lower average return and higher variance than the first strategy. To resolve this issue, I employ the modified Sharpe Ratio with X higher than the maximum absolute average return, say 0.3. The modified Sharpe ratios of the first, second and third strategy are 0.2, 0.14, and 0.10 respectively. The modified Sharpe ratio does not preserve the meaning of Sharpe ratio, but is a tool to compare risk-return profile of strategies effectively, implying which strategies bring more gains per unit of risk. In the earlier example, apparently strategy 1 creates more value than strategy 2 and 3, as correctly indicated with the highest modified Sharpe ratio. Column 4 in the table below reports the modified Sharpe Ratio of all the policies aforementioned.

	Mean	Variance	Modified Sharpe Ratio
Observed Optimal Policy	2.204		
Learning rate 0.1 (1)	-10.222	386.167	0.243
Learning rate 0.3 (2)	-10.877	381.059	0.211
Experience Replay with Learning Rate 0.1 – 0.1 (3)	-10.285	340.885	0.255
Experience Replay with Learning Rate 0.3 – 0.1 (4)	-10.268	360.479	0.249
Prioritized Experience Replay with Learning Rate 0.1 (5)	-11.105	402.486	0.194
Uniformly Random Experience Replay with Learning Rate 0.1 (6)	-10.760	359.52	0.224
Submit and Leave (7)	-10.671	329.193	0.238
TWAP (8)	-14.410	878.160	0.020

Table 8: Out-of- sample performance of the policies based on historical data, including mean accumulative rewards in one episode (column 2), variance of the different policies compared with the observed optimal policies (column 3) and their modified Sharpe ratio (column 4), for stock CATY. Row 3 and 4 report performance of policies learned from Q-learning with learning rate 0.1 and 0.3 respectively. Row 5 and 6 report performance of Q-learning policies with deterministic experience replay and learning rate 0.1-0.1 and 0.3-0.1 respectively. Row 7 and 8 show performance of Q-learning policies with random experience replay. Row 9 and 10 report performance of benchmark policies.

Observably, the optimal policies learned with learning rate 0.1 (model 1) outperform all of the remaining policies except for the ones learned with the deterministic experience replay (model 3 and model 4) in terms of modified Sharpe ratio. The fact that the policies in model 1 underperforms the policies in model 2 in both expected rewards and modified Sharpe ratio is consistent with expectation since learning rate 0.3 may be too large, making the estimates too volatile while learning rate 0.1 is small enough for the estimates to converge in the mean but still preserve the non-stationarity of the market. For more efficient use of data, I reutilize the training data with the same autocorrelation structure and train the agent with different learning rates for two running rounds. During the initial period of the training, the estimates can be volatile and updated quickly to catch up with the more recent and more relevant periods, so I set the learning rate 0.3 for the first round and 0.1 for the second round. Policies learned with learning rate 0.1 for both rounds (model 3) outperform policies learned with learning rate 0.3 for the first round and 0.1 for the second round (model 4) in variance and Sharpe ratio. The policies in model 3 and model 4 also are superior to the policies in model 1 since model 3 and model 4 utilize more data and therefore the estimates of action-state values Q are more likely to converge, incurring lower variance. With random experience replay, the policies perform much worse than the ones learned with deterministic experience replay in terms of Sharpe ratio due to their larger variance which may be incurred by stochastic feature of random sampling. Another possible explanation for the worse performance, even in terms of expected rewards, of random experience replay compared to that of deterministic replay may be the importance of temporal correlation of episodes in the training set that is preserved by deterministic replay but not by random replay. As mentioned in the earlier section, prioritized replay introduces more bias than uniformly random replay, thereby without bias correction such as importance sampling weight or any other methods, performance in model 5 is expectedly inferior to that in model 6. Most importantly, TWAP strategy performs worst among all the policies considered in terms of expected rewards, variance and accordingly modified Sharpe ratio. Policies in model 1, 3 and 4 surpass Submit and Leave strategy in terms of expected rewards and modified Sharpe ratio. Further details of performance over the episodes in the test set can be found in the figure 4.

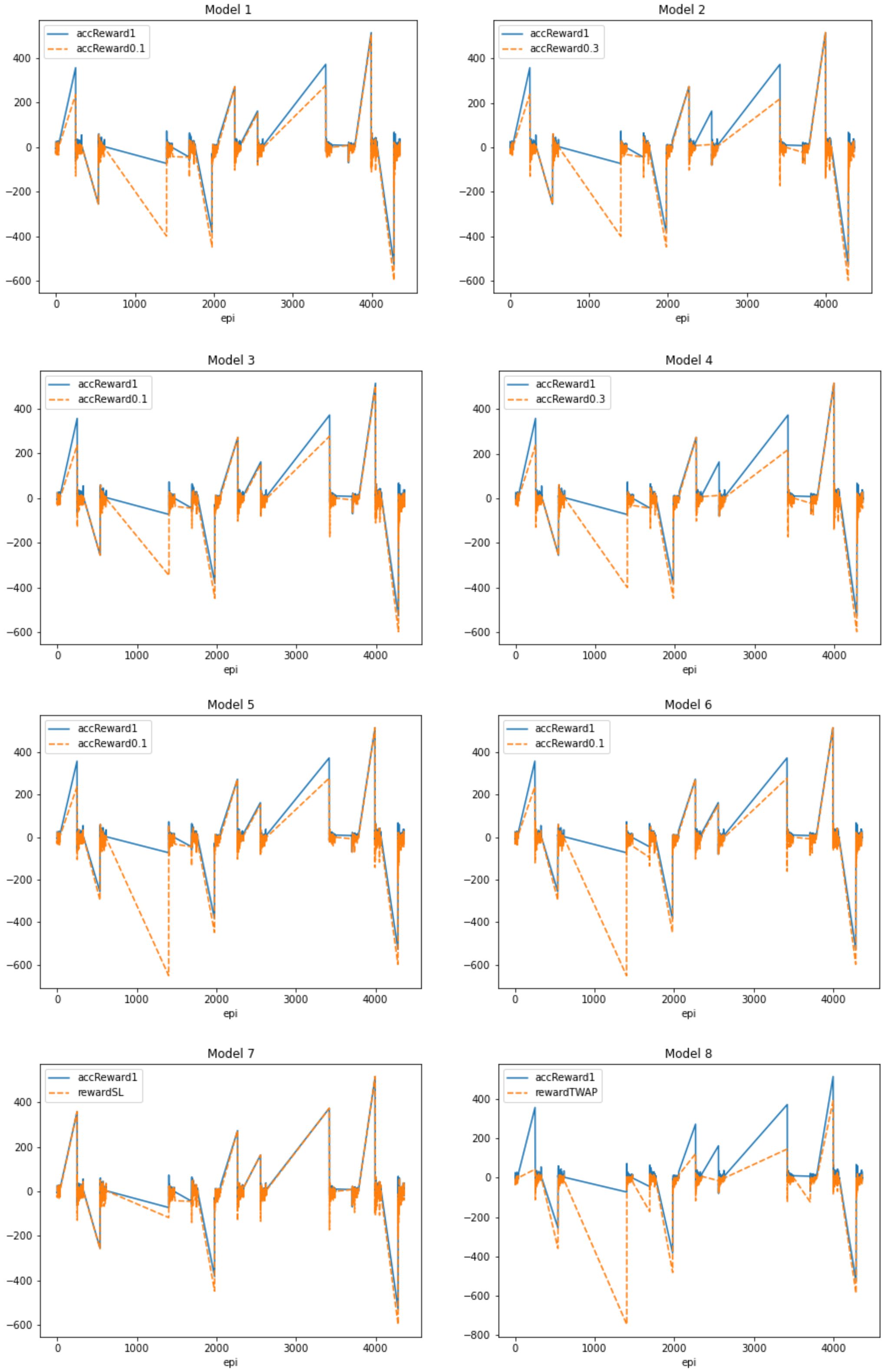

Figure 4: Performance of policies corresponding with models in table 8 relative to performance of observed optimal policies over the test set for stock CATY. accReward1 denotes accumulative reward of observed optimal policy in each episode.

The convergence of Q-learning requires that states be visited infinitely (Watskin and Dayan, 1992). However, this condition is rarely satisfied and we can approximate it if states are visited many times. As with a small dataset of this study, only more than 12.1% of states are visited more than 15 times, many state-action value estimates do not converge. Below are some illustrations of the convergence of one of the most often visited states (first column) and one of the less often visited states (second column) for model 1 (first row) and model 3 (second row). Since model 3 reutilize the training set, the starting Q estimates are from the previous training with learning rate 0.1, thereby different from 0 as shown in the second row. Observably, the estimates in model 3 appear to less volatile than those in model 1. This result explains the higher variance of policies in model 3 as compared to model 1. Additionally, though the state in the first column experiences a shock, it then gradually converges along with number of episodes. If it had been visited only 15 times, the state-action value estimate may have been much more volatile. This stresses the importance of a large number of visits to states.

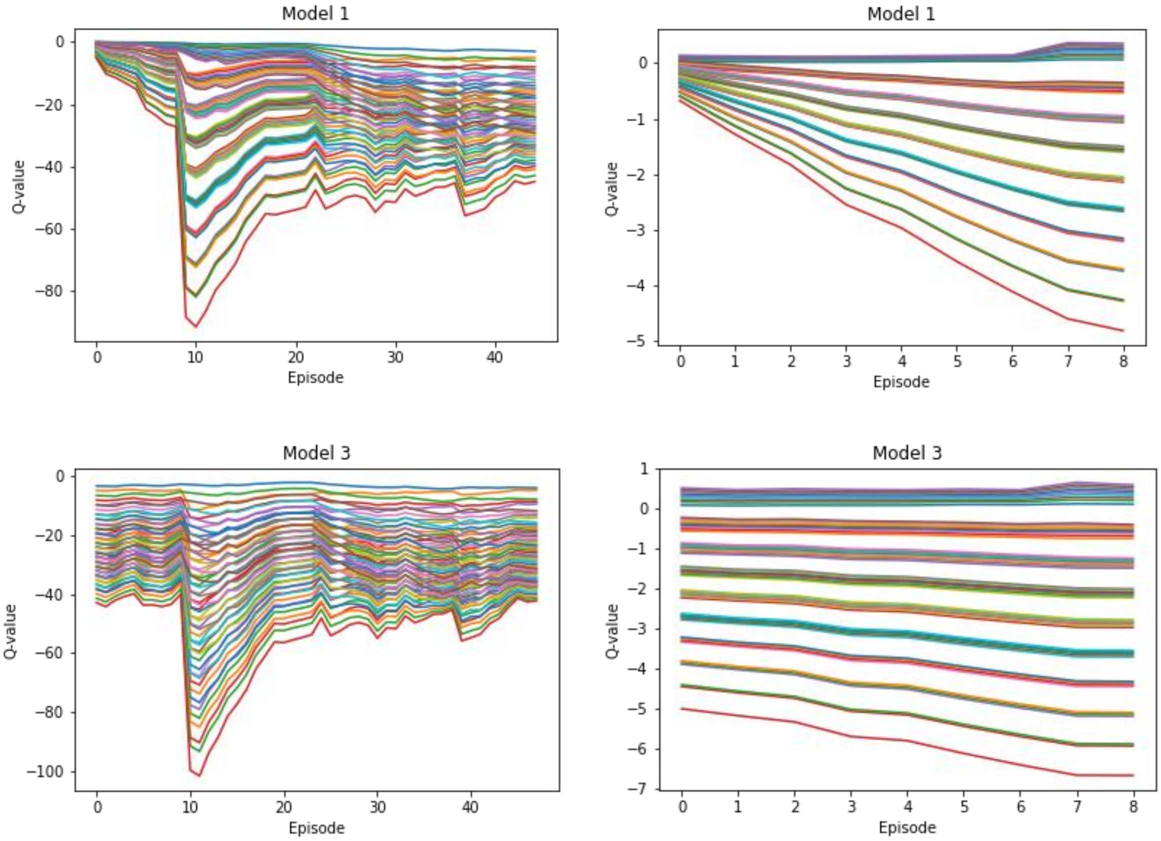

Figure 5: An illustration of convergence of two states in model 1 and model 3 in table 8 based on historical train dataset for stock CATY.

V. Limit Order Book Simulator

As mentioned in the earlier section, Q-learning and especially deep Q-learning require a large amount of data for the estimates to converge. For instance, deep Q-learning often consumes millions of observations to perform reasonably well. Sometimes, since the extensive collection of historical data is too expensive or not even accessible, researchers and practitioners often opt for simulated data. If they have the luxury to choose, they also need to consider the trade-off between use of huge amount of historical data and use of simulated data. Using a huge amount of historical data risks the chance that the market dynamics changes which may evolve quickly, thereby a large portion of the dataset likely to become irrelevant. Simulated LOB data on the other hand can be cheaply generated and generate scenarios that do not exist in training set but may occur in the test set, but may not very presentative of the true data due to modeling error. Despite enormous efforts of research in modeling LOB, there are still some gaps in explanation of the empirical properties of LOB. Below I present one of the most popular LOB simulators in the literature, namely Zero-Intelligence framework, which has been widely applied in deep Q-learning literature of this domain, and compare the performance of Q-learning using simulated data with that using historical one.

5.1. Simulation Algorithm – Zero-Intelligence Framework

Following Cont et al. (2010) and Abergel and Jedidi (2013), I assume initial limit order book is impacted by trades, limit orders and cancellations which arrive according to three independent Poisson processes. Arrival of trades of size ε^M follows Poisson process with intensity λ. Limit orders of size ε_i^L arrive at distance i ticks from the opposite best quote in accordance with Poisson process with intensity α_i. This model assumes that the simulator can generate an order that is at B absolute distance from the current opposite best quote at maximum. Cancellation of a limit order at distance i follows Poisson process with intensity $\delta_i \ a_i$ where a_i is the volume at a distance i from the opposite best quote. This essentially means each limit order at either bid side or ask side located at distance i from the opposite best quote has a life time drawn from the exponential distribution with intensity δ_i. This framework further assumes the size ε^M of market order, ε_i^L of limit order and ε_i^C of cancel order follow lognormal distributions, two signs of trades and limit orders are equally probable and there is no temporal correlation of orders. In short, the parameter set of the model is $(\lambda, \varepsilon^M, \alpha_1, \ldots, \alpha_B, \varepsilon_1^L, \ldots, \varepsilon_B^L, \delta_1, \ldots, \delta_B, \varepsilon_1^C, \ldots, \varepsilon_B^C)$. Below is the simulation algorithm of the Zero-Intelligence model.

```
Limit Order Book Simulator for an interval [0, T]
1. Initialization: set t = 0 and define A = 2 $\sum_{i=1}^{B} \alpha_i$
2. Compute $Na_{i,t}$ for all $1 \le t \le B$ as the number of outstanding shares in the ask side present at i ticks from the best bid quote $b_t$ at time t, $Nb_{i,t}$ for all $1 \le t \le B$ as the number of outstanding shares in the bid side present at i ticks from the best ask quote $a_t$ at time t.
3. Then compute the cancellation probability for limit ask orders and limit bid orders.
        $Na_t^* = \sum_{i=1}^{B} \delta_i \ Na_{i,t}$  and  $Nb_t^* = \sum_{i=1}^{B} \delta_i \ Nb_{i,t}$
4. Draw an event e = {1, 2, 3, 4} in their relative probabilities {2$\lambda$, A, $Na_t^*$, $Nb_t^*$}
5. If e = 1, generate a market order:
     • Randomly draw its sign
     • Draw its size from lognormal distribution with parameters $(\mu^M, \sigma^M)$
   If e = 2, generate a limit order:
     • Randomly draw its sign
     • Draw a limit order at a distant i ∈ $\{1, \ldots, B\}$from their relative probabilities $\{\alpha_1, \ldots, \alpha_B\}$
     • If its sign is -1, its limit price is $a_t - i, else\ its\ limit\ price\ is\ b_t + i$
     • Draw its size from lognormal distribution with parameters $(\mu_i^L\ , \sigma_i^L)$
   If e = 3, generate a cancellation at ask side:
     • Draw a distant i ∈ $1, \ldots, B$ from their relative probabilities $\{\delta_1 Na_{t,1}, \ldots, \delta_B Na_{t,B}\}$
     • Cancel a limit ask order at price $b_t + i$
     • Draw its size from lognormal distribution with parameters $(\mu_i^C\ , \sigma_i^C)$
   If e = 4, generate a cancellation at bid side:
     • Draw a distant i ∈ $\{1, \ldots, B\}$ from their relative probabilities $\{\delta_1 Nb_{t,1}, \ldots, \delta_B Nb_{t,B}\}$
     • Cancel a limit ask order at price $a_t - i$
     • Draw its size from lognormal distribution with parameters $(\mu_i^C\ , \sigma_i^C)$
6. Update Limit order book
```

7. Compute S = 2λ + A + Na_t^* + Nb_t^* and generate waiting time τ for the next event from an exponential distribution with a parameter S. Update t = τ + t

8. Repeat steps 2 to 7 until t > T

Calibration of Zero-intelligence Model

The set of parameter $(\lambda, \varepsilon^M, \alpha_1, \ldots, \alpha_B, \varepsilon_1^L, \ldots, \varepsilon_B^L, \delta_1, \ldots, \delta_B, \varepsilon_1^C, \ldots, \varepsilon_B^C)$ can be estimated from the historical order book over the period T as follows:

- $\hat{\lambda} = \frac{\#\, Trades}{2T}$
- $\widehat{\alpha_{\imath}} = \frac{\#\, Limit\ orders\ arrived\ at\ distance\ i\ from\ the\ opposite\ best\ quote}{2T}$
- $\widehat{\delta_{\imath}} = \frac{\#\, Cancellations\ occuring\ at\ a\ distance\ i\ from\ the\ opposite\ best\ quote}{2\, V_i\, T}$

  (Where V_i is the average volume at a distance i from the opposite best quote)

In this study, $\hat{\lambda}$, $\widehat{\alpha_{\imath}}$ and $\widehat{\delta_{\imath}}$ are averaged across 11 days of the training set as the final estimates. As for volumes, I estimate the parameters ε^M, ε_i^L and ε_i^C of the lognormal distributions for each trading day in the training set using Maximum Likelihood Method and average them to get the final estimates.

5.2. Summary Statistics

In this study, I estimate the parameters for quote distribution of 15 ticks around the best opposite quote. When fitting the lognormal distributions to the real data of volume, I also trim data outside of the 95% confidence interval to remove outliers or influential observations, in an attempt to make the estimation more representative to the real data. The parameters are estimated and shown as table 9. Observably, limit orders at a further distance from the opposite best quote is less likely to arrive but once they do, they have a larger size. For cancellations, the distance from the opposite best quote is increasing with both the likelihood of cancellation and the size of cancelled orders.

Tick	$\hat{\lambda}$	$\hat{\alpha}$	$\hat{\delta}$	$\widehat{\varepsilon^M}$	$\widehat{\varepsilon^L}$	$\widehat{\varepsilon^C}$
				(μ, σ)	(μ, σ)	(μ, σ)
1	0.0258	0.0247	4.1307	(2.7991, 1.7389)	(4.0606, 1.3354)	(4.1307, 1.5567)
2		0.0459	3.9950		(3.8638, 1.3655)	(3.9950, 1.2027)
3		0.0499	3.9538		(3.8892, 1.3399)	(3.9538, 1.2395)
4		0.0466	3.9728		(3.9629, 1.2874)	(3.9728, 1.2354)
5		0.0404	4.0497		(4.0316, 1.2450)	(4.0497, 1.1926)
6		0.0359	4.1521		(4.1364, 1.1357)	(4.1521, 1.1017)
7		0.0299	4.2163		(4.1996, 1.0778)	(4.2163, 1.0474)
8		0.0240	4.3000		(4.2663, 1.0309)	(4.3000, 0.9898)
9		0.0202	4.3119		(4.2977, 1.0077)	(4.3119, 0.9775)
10		0.0172	4.3639		(4.3429, 0.9532)	(4.3639, 0.9374)
11		0.0147	4.3844		(4.3692, 0.9757)	(4.3844, 0.9779)
12		0.0126	4.4107		(4.4329, 0.9233)	(4.4107, 0.9907)
13		0.0106	4.4584		(4.4793, 0.9283)	(4.4584, 0.9615)
14		0.0088	4.4960		(4.5145, 0.8906)	(4.4960, 0.9407)
15		0.0075	4.5805		(4.5830, 0.8949)	(4.5805, 0.9092)

Table 9: Estimated parameters of Zero-intelligence model for limit order book simulator for stock CATY. $\hat{\lambda}, \hat{\alpha}$ and $\hat{\delta}$ are estimated intensity parameter of Poisson processes for arrival of trades, limit orders and cancel orders respectively. $\widehat{\varepsilon^M}, \widehat{\varepsilon^L}$ and $\widehat{\varepsilon^C}$ are estimated parameters of lognormal distributions for size of trades, limit orders and cancel orders respectively.

Table 10 presents the summary statistics of simulated data. As compared to the summary statistics of actual data, all of the simulated key variables appear to be in a reasonable magnitude except for average market depth and spread. Similarly found in the original paper of Abergel and Jedidi (2013), this framework consistently underestimates spread probably due to the lack of consideration of temporal correlation of orders in the order book. However, as spread is one of the crucial factors to determine state space that is employed to train the agent, this underestimation may lead to the agent's failure in visiting the actual states that may occur in the test set and therefore in estimation of state-action Q values.

	Count	Mean	Standard Deviation	25% percentile	50% percentile	75% percentile
Mid-price change	25739	0	0.015	0	0	0
OFI	25740	0.651	472.483	-169	0	167
Average Market Depth	25740	218.471	198.078	-104.600	170	269.900
Spread	25740	0.014	0.026	0.01	0.01	0.02
Trading Volume	25740	53.496	85.404	2	21	69
Trading Imbalance	25740	0.634	86.301	-15	0	16
Order Imbalance	25740	2.688	403.312	-126	-1	127
Moving Difference	25740	0	0.017	-0.008	0	0.008

Table 10: Summary statistics of simulated data for stock CATY.

5.3. Performance Evaluation Based on Simulated Data

The agent is then trained on the simulated data with learning rate 0.1 and the performance of the learned policies is also examined with the test set. The state space comprises the same market and private variables as before, but all of the market variables are discretized based on 30% quantile and 70% quantile since their simulated data are more continuous. The table below shows the out-of-sample performance of the agent trained on the simulated data. Despite being learned on a substantially larger dataset, the policies underperform most of the policies learned from historical data and Submit and Leave strategy but still surpass TWAP strategy in terms of mean rewards and reward variance relative to the observed optimal policies. This analysis well serves as a warning for researchers or practitioners who solely rely on simulated data.

Mean	**Var**	**M. Sharpe Ratio**
-10.585	457.563	0.2064

Table 11: Out-of-sample performance of the policies learned from simulated data and with learning rate 0.1, including the mean accumulative rewards in one episode (column 1), variance of the different policies compared with the observed optimal policies (column 2) and their modified Sharpe ratio (column 3), for stock CATY.

VI. Robustness Test

6.1. Empirical Results Based on Historical Data

I further examine the performance of the model on 14 other stocks with different liquidity levels which are approximated as average trading volume of the stocks. The state space for 9 out of 14 stocks is defined similarly to that for stock CATY. For other stocks, due to their own properties, I select different market states based on the analysis of the observed optimal policies and market variables. Particularly, the chosen discretized market states for stock CBSH, BRKS and BJRI are discretized best bid volume, discretized trading imbalance, discretized moving difference, discretized spread, discretized order imbalance and discretized micro-price. The market states for stock AFYA and ACGL are similar except for best bid volume replaced with trading volume. For stock ACGL, due to its large trading volume, trading imbalance and trading volume are discretized by 30% and 70%. For stock CBSH, due to higher price volatility, moving difference is calculated based on 5 periods instead of 10. As stock AFYA and ANIP are much less often traded, I discretize trading imbalance by threshold 0 and 10. Table 12 shows the performance valuation of all the concerned policies for 15 stocks sorted by liquidity measure in a descending order.

All of the Q-learning policies outperform TWAP strategy and for 10 out of 15 stocks, Q-learning policies surpass Submit and Leave strategy in terms of modified Sharpe ratio. Among the 5 stocks, performance of Submit and Leave strategy is only marginally superior to that of Q-learning policies for AFYA, BRKS and BJRI but substantially exceeds that of Q-learning policies for ACGL and CBSH. As both ACGL and CBSH are highly liquid, their price changes may be more random and exposed to a variety of factors and their Q-learning models accordingly may require a larger state space to represent the environment. It should be emphasized that since I perform the analyses on the setting that market orders can always be executed at the best quote due to the data constraint as aforementioned, Submit and Leave strategies always outperform TWAP strategies in this test set. If I adjust the setting that only partial market orders are filled at the best quote, the impact on Submit and Leave strategy will be much more adverse, but on Q-learning policies less severe and on TWAP mild since Submit and Leave strategy usually involves placing a bulk market order at the end of episodes while Q-learning policies and TWAP strategy spread market orders over the whole episodes. In this case, the state space in Q-learning should be larger to better reflect the changes in limit

	Observed Optimal Policy	Q-Learning Policy with Learning Rate 0.1			Submit and Leave Strategy			TWAP Strategy		
	Mean	Mean	Var	M. SR	Mean	Var	M. SR	Mean	Var	M. SR
ACGL	**8.46**	**-4.15**	**486.58**	**0.084**	**-3.66**	**250.88**	**0.148**	**-5.11**	**1088.45**	**0.027**
HSIC	13.39	-9.19	2080.73	0.062	-9.55	2067.73	0.054	-11.08	1540.42	0.023
CRUS	20.08	-13.67	3865.84	0.054	-15.33	4923.45	0.024	-16.06	2099.35	0.020
CBSH	**18.99**	**-0.75**	**784.72**	**0.152**	**0.48**	**547.49**	**0.234**	**-4.47**	**1458.09**	**0.014**
BJRI	**18.84**	**-13.81**	**2716.57**	**0.080**	**-13.20**	**3092.21**	**0.086**	**-17.49**	**3397.69**	**0.009**
CVLT	8.66	-13.70	1295.99	0.147	-14.30	1309.58	0.130	-18.22	1246.50	0.022
BRKS	**6.53**	**-15.09**	**933.40**	**0.128**	**-15.38**	**776.19**	**0.130**	**-18.34**	**915.66**	**0.022**
JCOM	-6.55	-38.20	1698.27	0.116	-38.96	1781.98	0.096	-42.91	1534.75	0.002
CORE	-3.47	-17.39	322.28	0.201	-17.73	330.34	0.180	-20.66	268.29	0.021
CATY	2.204	-10.22	386.17	0.243	-10.67	329.19	0.238	-14.41	878.160	0.020
ALRM	2.73	-21.57	1206.09	0.127	-23.50	1405.00	0.067	-25.72	818.80	0.010
ALLK	-34.46	-91.46	7805.99	0.165	-96.62	11036.26	0.089	-105.56	8842.01	0.005
BOKF	-6.55	-41.00	3413.98	0.274	-47.72	4626.83	0.136	-56.97	4673.76	0.000
AFYA	**-53.08**	**-79.44**	**3267.96**	**0.080**	**-79.28**	**2853.57**	**0.088**	**-83.38**	**3359.13**	**0.011**
ANIP	-16.84	-58.13	3689.07	0.146	-58.69	4561.91	0.123	-66.75	2506.80	0.005

Table 12: Out-of-sample performance of Q-learning policies learned from historical data and with learning rate 0.1, Submit and Leave strategy and TWAP strategy for 15 stocks. Mean is the mean accumulative rewards in one episode. Var is variance of the different policies compared with the observed optimal policies. M. SR denotes modified Sharpe ratio.

order book, particularly deeper market depths. Most of the literature in this domain, however, considers neither market impacts of orders nor deeper market depth.

6.2. Empirical results Based on Simulated Data

I also extend the analysis on the 14 stocks based on simulated data. The state space for these 14 stocks hereafter is similar to that in the analysis based on historical data. However, only for much less traded stocks such as AFYA and ANIP, as simulated trading volume is less substantial than historical one, I discretize trading volume and trading imbalance by 0 and 10. Table 13 presents the performance evaluation of Q-learning policies learned from simulated data. Notably, Q-learning policies learned from simulated data also all outperform TWAP strategy in terms of modified Sharpe ratio but only surpass Q-learning policies learned from historical data for 3 out of 15 stocks. As these 3 stocks are all highly liquid, their price changes may be more exposed to stochastic properties and larger dataset may help the state-action value estimates to converge despite modeling error of LOB.

VII. Conclusion

This thesis has three main goals: first, to propose Q-learning algorithm for optimal trading execution with flexible execution strategies using two types of orders and various volume levels in the presence of market impact of limit and market orders; second, to present an empirical method to select state space for this optimal trading execution problem and third, to evaluate performance of Q-learning policies based on historical and simulated data relative to Submit and Leave strategy and TWAP strategy. Regarding the first goal, I design action space to comprise of limit orders and market orders of a variety of volumes, from 0 to the target volume of the parental order. The price impact of limit order and market order is incorporated into Q-learning algorithm, estimated as the price change due to the changes in order book. I perform an empirical analysis of price change per 30 seconds and find that order flow imbalance, order imbalance and trading imbalance complementarily and significantly explain the price movement every 30 seconds on average at 99% confidence level. Then I include the estimated price change due to changes in these three imbalance measures caused by the agent's orders into Q-learning algorithm as the temporary price impact incurred at the next immediate time step during an episode. This design allows the agent to penalize market

	Observed Optimal Policy	Q-Learning Policy Learned from Historical Data			Q-Learning Policy Learned from Simulated Data			TWAP Strategy		
	Mean	Mean	Var	M. SR	Mean	Var	M. SR	Mean	Var	M. SR
ACGL	8.46	-4.15	486.58	0.084	-4.54	998.32	0.046	-5.11	1088.45	0.027
HSIC	**13.39**	**-9.19**	**2080.73**	**0.062**	**-9.23**	**1977.89**	**0.062**	**-11.08**	**1540.42**	**0.023**
CRUS	**20.08**	**-13.67**	**3865.84**	**0.054**	**-13.09**	**2806.12**	**0.074**	**-16.06**	**2099.35**	**0.020**
CBSH	**18.99**	**-0.75**	**784.72**	**0.152**	**-0.79**	**665.53**	**0.163**	**-4.47**	**1458.09**	**0.014**
BJRI	18.84	-13.81	2716.57	0.080	-14.90	3864.27	0.050	-17.49	3397.69	0.009
CVLT	8.66	-13.70	1295.99	0.147	-13.82	1245.48	0.147	-18.22	1246.50	0.022
BRKS	6.53	-15.09	933.4	0.128	-16.33	1171.80	0.078	-18.34	915.66	0.022
JCOM	-6.55	-38.20	1698.27	0.116	-39.10	1777.43	0.093	-42.91	1534.75	0.002
CORE	-3.47	-17.39	322.28	0.201	-3.47	350.20	0.170	-20.66	268.29	0.021
CATY	2.204	-10.22	386.17	0.243	-10.59	457.56	0.206	-14.41	878.160	0.020
ALRM	2.73	-21.57	1206.09	0.127	-23.01	1308.69	0.083	-25.72	818.80	0.010
ALLK	-34.46	-91.46	7805.99	0.165	-96.48	10356.15	0.093	-105.56	8842.01	0.005
BOKF	-6.55	-41.00	3413.98	0.274	-47.09	5748.25	0.131	-56.97	4673.76	0.000
AFYA	-53.08	-79.44	3267.96	0.080	-80.32	3367.53	0.063	-83.38	3359.13	0.011
ANIP	-16.84	-58.13	3689.07	0.146	-58.72	4565.37	0.122	-66.75	2506.80	0.005

Table 13: Out-of-sample performance of Q-learning policies learned from historical data and with learning rate 0.1, Q-learning policies learned from simulated data and with learning rate 0.1 and TWAP strategy for 15 stocks. Mean is the mean accumulative rewards in one episode. Var is variance of the different policies compared with the observed optimal policies. M. SR denotes modified Sharpe ratio.

order more than limit order, larger volume of the same order type more than smaller volume of the same order type and continuous order placements more than distant order placements, which is consistent with the intuition. However, three critical assumptions of this framework are that market orders are always filled at the best quote, limit orders are always placed at the best quote and price impact is temporary. For the first two assumptions, it is easy to expand the action space to include deeper market depths, but this is equivalent to requirement of much larger state space for Q-learning and accordingly substantially bigger dataset which is not available for this study. The third assumption is critical in the sense that this study is primarily aimed at passive investors who do not trade on information advantage and therefore whose orders are less likely to create permanent price impacts. However, this study does not account for temporary price impacts of orders that are incurred at the subsequent time steps, not just the next immediate one. This consideration requires the assumption of the number of time steps at which the order placement will create price impacts and can be included into the algorithm as n-step Q-learning in future study.

I also propose an empirical method to choose state space for Q-learning in the problem of optimal trading execution. To my knowledge, most of the studies by default select some state space that they consider critical or do not clarify the selection process. In this study, I first allow the agent to learn the optimal policies assuming that the agent could observe the price trajectory of the whole episode. The agent picks the optimal policies based on price level and price impact caused by changes in order flow imbalance, trading imbalance and order imbalance due to order placement to maximize the total expected rewards. Then I perform an empirical analysis to select the most significant factors that drive the observed optimal policies using linear probability model. To take into account non-linear relations between the observed optimal policies and order book variables, I could engineer non-linear features of order book or use deep learning but for simplicity and due to the limited volume of data, I only employ linear probability model. Other non-linear models may provide with more features to be included into state space which may require an enormously larger dataset for the estimates to converge. Using linear probability model, I manage to select some of the important order book variables that significantly explain the observed optimal behavior, namely trading volume, trading balance, moving difference, spread, order imbalance and for some stocks, micro price and volume of best opposite quote. This analysis also implies that the performance of the model would be more superior if more variables are included into the state space and if there is a sufficient data volume for the states to be visited many times. Thus, in the future, I would like to explore the application of deep q-learning to capture substantially richer state space and therefore could alleviate the assumption of order placement and order fill at the best quote to address the optimal trading execution problem more realistically.

With regard to the third main goal of this research, I perform the evaluation of Q-learning policies learned from historical data and simulated data compared to two benchmark strategies, namely Submit and Leave and TWAP strategy. For 12 out of 15 stocks, Q-learning policies learned from historical data outperform those learned from simulated data but Q-learning policies regardless of data type surpass TWAP strategy. For 10 out of 15 stocks, Q-learning policies learned from historical data with step size 0.1 outperform Submit and Leave strategy but for 3 among the remaining 5 stocks, performance of Submit and Leave strategy is only marginally superior to that of Q-learning policies. However, the relative performance of Q-learning policies, TWAP strategy and Submit and Leave strategy is affected by the assumption of order placement and order fill at the best quote. Allowing for deeper market depths may exacerbate the performance of Submit and Leave strategy most significantly. As mentioned earlier, simulated data is often employed in deep Q-learning as deep Q-learning requires a huge volume of data. Thus, due to the trade-off between historical data and simulated data, especially of a large volume, it is interesting to compare the performance of deep Q-learning using historical with that using simulated data in future work. I also perform the detailed analysis of experience replay on stock CATY to make more efficient use of data and find that experience replay helps to reduce the variance of the state-action value estimates and that deterministic experience replay may be more useful than random experience replay since it preserves the temporal correlations of episodes and accordingly may be more suitable in non-stationary environment. As random experience replay is a popular remedy to satisfy the independent and identical distribution assumption in function approximation such as deep Q-learning, it is helpful to evaluate the performance of Q-learning vs. deep Q-learning using random experience replay in future work.

BIBLIOGRAPHY

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

Agliardi, R., & Gençay, R. (2017). Optimal trading strategies with limit orders. *International Journal of Theoretical and Applied Finance*, *20*(01), 1750005.

Alfonsi, A., Fruth, A., & Schied, A. (2010). Optimal execution strategies in limit order books with general shape functions. *Quantitative Finance*, *10*(2), 143-157.

Almgren, R. F. (2003). Optimal execution with nonlinear impact functions and trading-enhanced risk. *Applied Mathematical Finance*, *10*(1), 1-18.

Almgren, R., & Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, *3*, 5-40.

Avellaneda, M., & Stoikov, S. (2008). High-frequency trading in a limit order book. *Quantitative Finance*, *8*(3), 217-224.

Bacry, E., & Muzy, J. F. (2014). Hawkes model for price and trades high-frequency dynamics. *Quantitative Finance*, *14*(7), 1147-1166.

Bacry, E., Delattre, S., Hoffmann, M., & Muzy, J. F. (2013). Some limit theorems for Hawkes processes and application to financial statistics. *Stochastic Processes and their Applications*, *123*(7), 2475-2499.

Bacry, E., Jaisson, T., & Muzy, J. F. (2016). Estimation of slowly decreasing hawkes kernels: application to high-frequency order book dynamics. *Quantitative Finance*, *16*(8), 1179-1201.

Bertsimas, D., & Lo, A. W. (1998). Optimal control of execution costs. *Journal of Financial Markets*, *1*(1), 1-50.

Bouchaud, J. P. (2009). Price impact. *arXiv preprint arXiv:0903.2428*.

Bouchaud, J. P. (2010). The endogenous dynamics of markets: price impact and feedback loops. *arXiv preprint arXiv:1009.2928*.

Bouchaud, J. P., & Potters, M. (2003). *Theory of financial risk and derivative pricing: from statistical physics to risk management*. Cambridge University Press.

Cartea, A., & Jaimungal, S. (2013). Modelling asset prices for algorithmic and high-frequency trading. *Applied Mathematical Finance*, *20*(6), 512-547.

Cartea, A., & Jaimungal, S. (2015). Optimal execution with limit and market orders. *Quantitative Finance*, *15*(8), 1279-1291.

Chakraborti, A., Toke, I. M., Patriarca, M., & Abergel, F. (2011). Econophysics review: I. Empirical facts. *Quantitative Finance*, *11*(7), 991-1012.

Cont, R., & Kukanov, A. (2017). Optimal order placement in limit order markets. *Quantitative Finance*, *17*(1), 21-39.

Cont, R., Kukanov, A., & Stoikov, S. (2014). The price impact of order book events. *Journal of Financial Econometrics*, *12*(1), 47-88.

Cont, R., Stoikov, S., & Talreja, R. (2010). A stochastic model for order book dynamics. *Operations Research*, *58*(3), 549-563.

Dempster, M. A., & Leemans, V. (2006). An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications*, *30*(3), 543-552.

Engle, R. F., & Patton, A. J. (2004). Impacts of trades in an error-correction model of quote prices. *Journal of Financial Markets*, *7*(1), 1-25.

Evans, M. D., & Lyons, R. K. (2002). Order flow and exchange rate dynamics. *Journal of Political Economy*, *110*(1), 170-180.

Forsyth, P. A., Kennedy, J. S., Tse, S. T., & Windcliff, H. (2012). Optimal trade execution: a mean quadratic variation approach. *Journal of Economic dynamics and Control*, *36*(12), 1971-1991.

Fruth, A., Schöneborn, T., & Urusov, M. (2019). Optimal trade execution in order books with stochastic liquidity. *Mathematical Finance*, *29*(2), 507-541.

Gandhi, N., Allard, M., Kim, S., Kazanzides, P., & Bell, M. A. L. (2017). Photoacoustic-based approach to surgical guidance performed with and without a da Vinci robot. *Journal of Biomedical Optics*, *22*(12), 121606.

Gatheral, J. (2010). No-dynamic-arbitrage and market impact. *Quantitative finance*, *10*(7), 749-759.

Guéant, O., Lehalle, C. A., & Fernandez-Tapia, J. (2012). Optimal portfolio liquidation with limit orders. *SIAM Journal on Financial Mathematics*, *3*(1), 740-764.

Guéant, O., Pu, J., & Royer, G. (2015). Accelerated Share Repurchase: pricing and execution strategy. *International Journal of Theoretical and Applied Finance*, *18*(03), 1550019.

Guilbaud, F., & Pham, H. (2013). Optimal high-frequency trading with limit and market orders. *Quantitative Finance*, *13*(1), 79-94.

Hasbrouck, J., & Seppi, D. J. (2001). Common factors in prices, order flows, and liquidity. *Journal of Financial Economics*, *59*(3), 383-411.

Hautsch, N., & Huang, R. (2012). The market impact of a limit order. *Journal of Economic Dynamics and Control*, *36*(4), 501-522.

Hendricks, D., & Wilcox, D. (2014, March). A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution. In *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)* (pp. 457-464). IEEE.

Hopman, C. (2007). Do supply and demand drive stock prices?. *Quantitative Finance*, *7*(1), 37-53.

Huberman, G., & Stanzl, W. (2004). Price manipulation and quasi-arbitrage. *Econometrica*, *72*(4), 1247-1275.

Huitema, R. (2014). Optimal portfolio execution using market and limit orders. *Available at SSRN 1977553*.

Jaimungal, S., Kinzebulatov, D., & Rubisov, D. (2013). Optimal accelerated share repurchase. *Available at SSRN 2360394*.

Karpe, M., Fang, J., Ma, Z., & Wang, C. (2020). Multi-agent reinforcement learning in a realistic limit order book market simulation. *arXiv preprint arXiv:2006.05574*.

Karpoff, J. M. (1987). The relation between price changes and trading volume: A survey. *Journal of Financial and Quantitative Analysis*, *22*(1), 109-126.

Kempf, A., & Korn, O. (1999). Market depth and order size. *Journal of Financial Markets*, *2*(1), 29-48.

Kyle, A. S. (1985). Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society*, 1315-1335.

Lin, L. J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, *8*(3-4), 293-321.

Lin, S., & Beling, P. A. (2020, September). A Deep Reinforcement Learning Framework for Optimal Trade Execution. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 223-240). Springer, Cham.

Lu, X., & Abergel, F. (2017). Limit order book modelling with high dimensional Hawkes processes.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, *17*(5-6), 441-470.

Nevmyvaka, Y., Feng, Y., & Kearns, M. (2006, June). Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd international conference on Machine learning* (pp. 673-680).

Ning, B., Lin, F. H. T., & Jaimungal, S. (2018). Double deep q-learning for optimal execution. *arXiv preprint arXiv:1812.06600*.

Potters, M., & Bouchaud, J. P. (2003). More statistical properties of order books and price impact. *Physica A: Statistical Mechanics and its Applications*, *324*(1-2), 133-140.

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Schvartzman, L. J., & Wellman, M. P. (2009). Learning improved entertainment trading strategies for the tac travel game. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets* (pp. 195-210). Springer, Berlin, Heidelberg.

Shelton, C. R. (2001). Importance sampling for reinforcement learning with multiple objectives.

Smith, E., Farmer, J. D., Gillemot, L., & Krishnamurthy, S. (2003). Statistical theory of the continuous double auction. *Quantitative Finance*, *3*(6), 481.

Spooner, T., Fearnley, J., Savani, R., & Koukorinis, A. (2018). Market making via reinforcement learning. *arXiv preprint arXiv:1804.04216*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Vaes, J., & Hauser, R. (2018). Optimal execution strategy with an uncertain volume target. *arXiv preprint arXiv:1810.11454*.

Vyetrenko, S., & Xu, S. (2019). Risk-sensitive compact decision trees for autonomous execution in presence of simulated market response. *arXiv preprint arXiv:1906.02312*.

Watkins, C. J. C. H. (1989). Learning from delayed rewards.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279-292.

Weber, P., & Rosenow*, B. (2005). Order book approach to price impact. *Quantitative Finance*, *5*(4), 357-364.

University of Mannheim | Business School
Chair of Finance
Prof. Dr. Erik Theissen

UNIVERSITY
OF MANNHEIM
Business School

Eidesstattliche Erklärung

"Hiermit versichere ich, dass diese Abschlussarbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen. Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann."

Nguyen Thi Hoa
Mannheim, 4 September 2021

Ort, Datum, eigenhändige Unterschrift