Market Making under Order Stacking Framework: A Deep Reinforcement Learning Approach

Guhyuk Chung
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
wjdrngur12@kaist.ac.kr

Munki Chung
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
moonki93@kaist.ac.kr

Yongjae Lee*
Ulsan National Institute of Science and Technology
Ulsan, Republic of Korea
yongjaelee@unist.ac.kr

Woo Chang Kim*
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
wkim@kaist.ac.kr

ABSTRACT

Market making strategy is one of the most popular high frequency trading strategies, where a market maker continuously quotes on both bid and ask side of the limit order book to profit from capturing bid-ask spread and to provide liquidity to the market. A market maker should consider three types of risk: 1) inventory risk, 2) adverse selection risk, and 3) non-execution risk. While there have been a lot of studies on market making via deep reinforcement learning, most of them focus on the first risk. However, in highly competitive markets, the latter two risks are very important to make stable profit from market making. For better control of the latter two risks, it is important to reserve good queue position of their resting limit orders. For this purpose, practitioners frequently adopt order stacking framework where their limit orders are quoted at multiple price levels beyond the best limit price. To the best of our knowledge, there have been no studies that adopt order stacking framework for market making. In this regard, we develop a deep reinforcement learning model for market making under order stacking framework. We use a modified state representation to efficiently encode the queue positions of the resting limit orders. We conduct comprehensive ablation study to show that by utilizing deep reinforcement learning, a market making agent under order stacking framework successfully learns to improve the P&L while reducing various risks. For the training and testing of our model, we use complete limit order book data of KOSPI200 Index Futures from November 1, 2019 to January 31, 2020 which is comprised of 61 trading days.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

* Corresponding authors.

ICAIF '22, November 2–4, 2022, New York, NY, USA

ACM ISBN 978-1-4503-9376-8/22/11…$15.00
https://doi.org/10.1145/3533271.3561789

KEYWORDS

market making, order stacking, high-frequency trading, market microstructure, deep reinforcement learning

ACM Reference Format:
Guhyuk Chung, Munki Chung, Yongjae Lee, and Woo Chang Kim. 2022. Market Making under Order Stacking Framework: A Deep Reinforcement Learning Approach. In *3rd ACM International Conference on AI in Finance (ICAIF '22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3533271.3561789

1 INTRODUCTION

Market making is to continuously quote on both bid and ask side of a limit order book to make profit from capturing bid-ask spread while providing liquidity to the market. There are three sources of risk in market making strategies: 1) inventory risk, 2) adverse selection risk, and 3) non-execution risk. First, inventory risk is the risk of price moving in unfavourable direction when the market maker's inventory is skewed to one side. This is the major risk specific to market making, because market making strategies aim to keep the inventory level close to zero and focus on making profit from bid-ask spread. Note that speculation strategies, which is another popular high-frequency trading scheme, would not care about skewed inventory because they aim to profit from predicting short-term price movements. Second, adverse selection risk is the risk of best bid (ask) price moving down (up) shortly after the market maker's limit bid (ask) order is executed. This is unfavourable because if the best bid (ask) price would have gone down (up), it might have been better to just cancel the bid (ask) order before being executed and wait at the next bid (ask) price level. Third, non-execution risk is the risk of best bid (ask) price moving up before the market maker's limit bid (ask) order at the best price level gets executed. If the best bid (ask) price would have gone up (down), the market maker could have rather chosen to take the liquidity of the ask (bid) side using market order before it gets depleted. Figure 1 shows the cases of adverse selection (above) and non-execution (below) of one single bid order.

Among the three sources of risks in market making, adverse selection risk and non-execution risk are closely related to the queue position of a limit order. Queue position here refers to the position of an order inside the queue and indicates whether the order is relatively ahead or behind. It is clear that if the queue is long and the order is located relatively at the front of the queue,

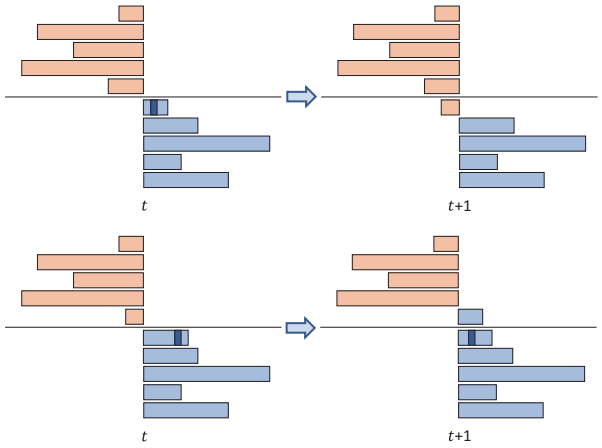

Figure 1: Limit order book snapshots at time step t and $t+1$. The highlighted box represents the queue position of the market maker's order. Above: Adverse selection of a bid order. Below: Non-execution of a bid order.

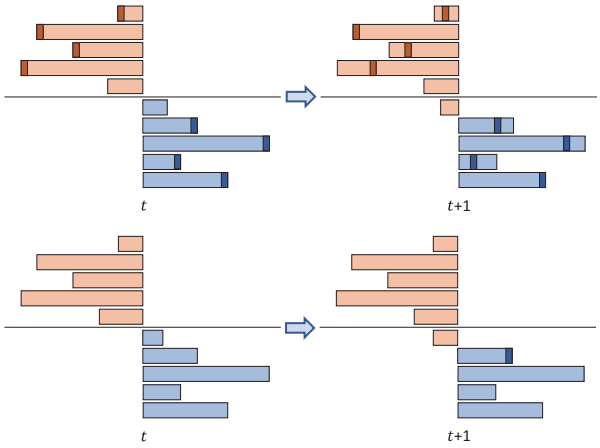

Figure 2: Comparison between order stacking (above) and no order stacking (below). The highlighted boxes represent the queue positions of market maker's orders.

both adverse selection risk and non-execution risk would be low. If the order is located relatively behind the queue and the queue is long, non-execution risk would be high, and if the queue is short adverse selection risk would be high. Most of the market making literature focus less on these two sources of risk compared to the inventory risk in the sense that they do not explicitly control these two risks by utilizing queue position information. [12] tracks the queue position of the orders during simulation, but only uses this information to check whether the order would be executed and does not include this information in the state space. [4] implicitly assumes that market maker's order is located at the front of the queue as soon as the order is submitted by adopting first-passage time execution model described in [13]. However, this assumption is too optimistic for most market environments.

Reserving good queue positions beyond the best bid/ask price levels is desirable in many aspects including better control of adverse selection risk and non-execution risk. Thus, practitioners frequently deploy order stacking strategies that place limit orders at multiple price levels in advance. Figure 2 shows how order stacking can be helpful in obtaining better queue positions. Assume at time step t, the market maker quotes one limit order at each level beyond the best price level for both bid and ask side. All of these orders are at the end of the queue at the time of submission. From t to $t+1$, orders in front of the market maker's orders can be canceled and new orders from other market participants can be submitted following the market maker's orders. If the orders at the best bid is depleted during this time period, best bid price moves down. Under order stacking framework, the resting limit order at the new best bid price would be in the middle of the queue (above), whereas without order stacking, the newly submitted limit order would be at the end of the queue (below).

In case of highly competitive and liquid assets such as KOSPI200 Index Futures, it is extremely rare that one large market order taking the entire liquidty at the best bid (or best ask) and changing the midprice accordingly. This means that resting limit orders beyond the best level have very low risk of being executed, thus there is no reason for the market maker to cancel these stacked orders until they are located at the best price level. Hence it is sufficient for the market maker to stack orders at multiple price levels and only care about the best price level, i.e., whether to insert, stay, or cancel the limit orders at the best bid/ask.

In this paper, we develop a market making strategy under order stacking framework using a deep reinforcement learning approach to better control adverse selection risk and non-execution risk in highly competitive markets. For highly competitive markets such as KOSPI200 Index Futures, market participants mostly place limit orders, which would increase non-execution risk for newly placed orders and adverse selection risk for executed orders. We show that order stacking can be helpful in making stable profit in such difficult market environment. In addition, we use model-free deep reinforcement learning technique to find optimal market making strategies while considering multiple queue positions, inventory, and market dynamics at the same time.

Our contributions in this work are as follow. First, to the best of our knowledge, this is the first work to establish market making problem under order stacking framework and solve it via reinforcement learning. Here we include *Queue Position Vector, QPV* to the state space to efficiently encode the information of multiple queue positions resting at different levels of the limit order book. Second, we provide carefully designed implementation details of both LOB simulation environment and deep reinforcement learning agent. Finally, we report comprehensive empirical results with 61 days of KOSPI200 Index Futures LOB data. Especially, we focus not only on the improvement of P&L, but also on the decrement of adverse selection. We show that introducing order stacking framework, utilizing deep reinforcement learning technique, and using QPV for multiple queue positions are all helpful for achieving better P&L and reducing adverse selection risk.

1.1 Related work

Finding optimal market making strategies has traditionally been studied under stochastic optimal control framework since the seminal work of [1]. Here, the dynamics of limit order book is explicitly modeled by modeling the order arrivals with stochastic processes. This line of research focuses on controlling inventory risk and explicitly penalizes holding inventory by adding the penalizing term in the objective function.

More recently, model-free reinforcement learning approach has gained popularity after the work of [12]. They used the high frequency limit order book data of 8 months with 10 securities for simulation, which is one of the largest datasets for high-frequency trading studies. The linear combination of tile codings was used for efficient representation of state space, and various value-based reinforcement learning algorithms were tested. The P&L function was divided into three separated terms, where the first two terms capture the P&L from bid-ask spread and the last term capture the P&L from price movements while holding inventory. They tried to prevent holding inventory in market making by dampening the last term. [4] suggested to include prediction signals of future trend and volatility in the state space by pre-training a deep learning model and gradient-boosting model. They also suggested a linear reward function that directly penalizes holding inventory. [14] developed a novel market feature named *Book Exhaustion Rate, BER* and showed that including this feature in the state space leads to better adverse selection risk control. However, they used different definition of adverse selection risk from ours. They defined adverse selection as the case when the bid (ask) order gets executed, price moves down (up), and never comes back to the executed price level until the end of trading horizon. Our definition of adverse selection risk and non-execution risk is based on the work of [3, 5, 8] and references therein.

There are studies that explicitly address adverse selection risk and non-execution risk using the information of queue position and queue imbalance defined in equation (2). However, they do not consider market making problems, but their objective is to execute a *single* limit order at the most favourable price by explicitly controlling the two mentioned risks. Limit order book dynamics is modeled using stochastic processes, and the optimal limit order placement strategy is learned using stochastic control approach. [10] is one of the first works to develop a model for valuing an order conditioned on queue position. [8] and [5] both study optimal limit order placement strategy under adverse selection risk, while the latter work additionally controls non-execution risk by including market order types into the framework. [3] develops an analytic order scoring model based on queue position and queue imbalance, and uses multi-armed bandit approach to decide whether or not to cancel the resting limit order with the objective of minimizing adverse selection. Research on including multiple orders in this framework is scarce (if exists) because analytic solution to optimal control policy considering queue positions of multiple orders is hard to be obtained due to the high complexity of the model. One research that uses similar approach considering multiple orders and queue imbalance is [2] but they do not include queue positions of the orders into their model.

2 SIMULATION

In this section, we provide details of limit order book simulation in our work. In general, we follow the usual setup of other works [3, 12] , but we make some modifications for more realistic simulation.

2.1 Order Submission and Tracking

As in most previous studies on market making via reinforcement learning, we assume that every order submitted by market making agent is of unit quantity, and that the price impact is negligible. While most of the other papers allow up to one order for each side of the limit order book, we let up to one order for each price level. This is a natural difference of our work from others because we study market making under order stacking framework. For clarity and ease of expression, for the rest of the paper we assume every order resting in the limit order book is also of unit quantity. Then, for example, if the total volume of the orders at the best bid price is V, we say there are V orders resting at the best bid price.

We assume the newly submitted order of our agent starts from the end of the queue. We denote $Q^{bid,k}_{front}$ ($Q^{ask,k}_{front}$) as the number of orders in front of our order resting at the kth price level of bid (ask) side, and similarly denote $Q^{bid,k}_{behind}$ ($Q^{ask,k}_{behind}$) as the number of orders behind our order. We define the *Relative Queue Position, RQP* of our order resting at the kth price level of bid side as follows

$$RQP_{bid,k} = \frac{Q^{bid,k}_{front}}{Q^{bid,k}_{front} + Q^{bid,k}_{behind}} \tag{1}$$

which indicates the position of our order inside the queue, and $RQP_{ask,k}$ is defined in the same way. This value ranges from zero (front of the queue) to one (end of the queue). When a new limit order arrives, it is added to Q_{behind}, and when a new market order arrives, it is subtracted from Q_{front}. This is trivial, and no assumption is required. However, when a new cancel order arrives our data does not tell us the exact location of the canceled order so there is ambiguity as to whether this should be subtracted from Q_{front} or Q_{behind}. In the following subsection, we describe how we deal with this ambiguity.

2.2 Cancellation Assumption

To deal with the aforementioned ambiguity, [12] made the assumption that the canceled orders are uniformly distributed throughout the queue. Many other following works adopt this assumption. Motivated by this assumption, we have experimented with three different implementation details. The first option we call deterministic assumption is to subtract $q \cdot RQP$ from Q_{front} and subtract $q \cdot (1 - RQP)$ from Q_{behind} when the canceled quantity is q. The second option we call stochastic assumption is to subtract q from Q_{front} with probability of RQP and subtract q from Q_{behind} with probability of $1 - RQP$. The last option we call binomial assumption is to sample a value $x \sim Binomial(q, RQP)$, subtract x from Q_{front} and subtract $q - x$ from Q_{behind}. It is worth noting that when RQP is close to zero, there is high probability that zero quantity is subtracted from Q_{front} when following the second or last assumption, whereas certain amount of quantity is always subtracted from Q_{front} whenever cancel order arrives under the first assumption. We find that the first assumption is highly optimistic

for the data of KOSPI200 Index Futures we used in this work where the proportion of cancel orders is large. From our experiments, the performances under the second and last assumption are similar. All of the experiment results presented in Section 4 and Section 5 are obtained under the binomial assumption.

2.3 Execution Assumption

There are two cases where our resting order at the best bid/ask price is assumed to be executed. The first case is when a new sell (buy) market order of quantity larger than or equal to $Q_{front}^{bid,1}$ ($Q_{front}^{ask,1}$) arrives. This first case assumption is shared with many other papers and is a realistic assumption since we assume our order is of unit quantity. The second case could be more controversial. There can be cases when the volume at the best bid (ask) price is depleted without executions, i.e., all orders are canceled rather than being taken by opposite market orders, and the best bid (ask) price moves down (up). [3] assumes that if the best bid (ask) price moves down (up) without executions, our bid (ask) limit order resting at the previous best bid (ask) price is assumed to be not executed until an execution occurs at price lower (higher) than or equal to the price of our limit order. In contrast, we make a simpler assumption that our bid (ask) limit order at the best bid (ask) price is executed as soon as the best bid (ask) price moves down (up) even if there had been no market orders to execute our order.

3 METHODS

In this section, we describe our reinforcement learning agent for market making under order stacking framework. While the overall structure is similar to existing models [4, 12, 15], we formalize the order stacking framework (Section 3.3), suggest a novel state space definition to efficiently incorporate multiple queue position information (Section 3.4), and provide rationale and insights for designing the details.

3.1 Step Interval

There are two motivations that prevent the market making agent from making actions at every limit order book change. In this work and in many other relevant papers, zero latency is assumed both for order information retrieval from the exchange and new order submission. This assumption is too strong and naive if we assume the agent can react to every change of the limit order book since time between two consecutive orders in a highly liquid asset like KOSPI200 futures can be as short as magnitude of microsecond or shorter. Second, consecutive limit order book status are highly correlated, and these correlated samples would harm the reinforcement learning performance. To alleviate the problem of highly correlated consecutive states, frame skipping technique is widely used for learning Atari games [9]. In a similar manner, many works choose to make action once every predefined step interval. Step interval can be defined using physical time, i.e., second, minute, etc., or using number of fixed limit order book changes.

Motivated by the work of [4], we define an *event* as a limit order book change where the change is made at the top of the limit order book, i.e., the price/volume at the best price level has changed. Followed by this definition, we define one step interval as 50 events which corresponds to about 1~2 seconds on average during the period of our dataset for KOSPI200 Index Futures. There are two things to be mentioned after our definition of step interval. First, we do not use physical time for our step interval definition because number of limit order book changes during one step interval may vary significantly depending on the degree of market activation. It is well known that market is more active during time closer to market open/close and less active during mid-hours. Second, by only counting *events* for step interval, we can contain consistent information value. There would be little controversy to the claim that changes at the top of the limit order book is generally more important than changes deeper in the limit order book. If the step interval is defined using every change of limit order book, some intervals might contain large portion of *events* while other intervals might contain small portion of *events*, leading to inconsistency of information value.

3.2 Action Space

We adopt the action space definition of [15]. They define action space as $a = (action_{bid}, action_{ask}) \in \{(0,0), (0,1), (1,0), (1,1)\}$ where 1 indicates that the agent wants her order to rest in the best bid/ask price, and 0 indicates the agent wants no order resting in the best bid/ask price. Following this definition, the *actual* action that the agent should perform is different depending on whether the agent's order is already on the best bid/ask or not. For example, if the agent's order is already on the best bid, $action_{bid} = 1$ would mean that the agent does nothing, and $action_{bid} = 0$ would mean that the agent cancels her order at the best bid. Conversely, if the agent has no order at the best bid, $action_{bid} = 1$ would mean that the agent submits new limit order at the best bid, and $action_{bid} = 0$ would mean the agent does nothing.

3.3 Order Stacking

Note that most details regarding how the order stacking framework works and the benefits of order stacking are described in Introduction, and the modeling of action space under order stacking framework is described in the previous subsection. Here, we provide some additional implementation details for order stacking that are worth noting.

We restrict up to one limit order for each price level meaning that if one limit order already exists at certain price level, we consider that price level as *full* and does not submit additional order at that level. If no limit order exists at certain price level, we consider it as *empty*. The basic principle is to confirm that every price level up to 5th level for both sides is *full* except for the best price level which is controlled by the market making agent. We call this confirmation process *order filling*. Order filling is done every action step interval, i.e., 50 events, not every time the limit order book changes, to be coherent with the market making agent. We describe the *order filling* details with the focus on the bid side. Let's assume the best bid price goes up by one tick during time period from t to $t+1$. Then, the best bid price level at t would now become the second best price level. This price level is either *full* or *empty* depending on the action chosen by the agent at time step t and whether execution occurred during the time period. If it is *empty*, new limit order is submitted to this price level at $t+1$ and do nothing if it is *full*. In the meanwhile, the 5th price level at t would now become 6th price

level at $t+1$. Since our limit order book data provides up to 5th price level and thus we cannot track the queue position of the order at this price level any more, we assume this order is immediately canceled at $t+1$. Now assume the best bid price moves down by one tick during time period from $t+1$ to $t+2$. 6th price level at $t+1$ would now again become 5th price level at $t+2$, and since we assume the order at this level had been canceled at $t+1$, this level would certainly be *empty*. Thus we submit new limit order to this price level. The second best price level at $t+1$ would now become the best price level at $t+2$, and it would certainly be *full*.

3.4 State Space

Our state space definition is comprised of three components. The first element is responsible for representing market state, and the latter two are for representing agent state. To represent the market state, we use single scalar value *Queue Imbalance, QI* which is defined as follow

$$Queue\ Imbalance = \frac{Q^{bid,1} - Q^{ask,1}}{Q^{bid,1} + Q^{ask,1}} \tag{2}$$

where $Q^{bid,1}$ ($Q^{ask,1}$) represent the order quantity resting at best bid (ask) price level. While a single value seems insufficient to represent the complex dynamics of limit order book, many studies [2, 3, 6] claim that *QI* alone is a powerful predictor of next mid-price change direction. From the definition, the value of *QI* ranges from -1 to 1. When *QI* is close to -1, it is considered *sell-heavy*, and the price is more likely to go down. Likewise, when its value is close to 1, it is considered *buy-heavy* and the price is more likely to go up. Being aware of future price dynamics should help market making agent in various ways. For example, if the agent has its order resting at the best bid level and the mid-price is likely to go down, the agent would probably want to cancel its order to avoid adverse selection.

The second element is the current inventory level of the agent. For proper control of inventory risk, knowing the current inventory level is indispensable. If the inventory level is too skewed to one side, the market making agent might have more incentive to execute the opposite limit order even at the risk of adverse selection.

The last element is *Queue Position Vector, QPV*, which is a 10-dimensional vector of which each component encodes the queue position information at each bid/ask price level. The definition of *QPV* for each bid/ask price level is defined as follow

$$QPV_k = \begin{cases} 1 - RQP_k & \text{if } k\text{-th price level is } full \\ 0 & \text{if } k\text{-th price level is } empty \end{cases} \tag{3}$$

Note that $1 - RQP_k$ is well defined variable because from the definition of *RQP* in equation (1), $1 - RQP_k$ equals one means our order is at the front of the queue and $1 - RQP_k$ equals zero means either our order is at the end of the queue or our order is not placed in the queue. Having no order in the queue and having order at the end of the queue can roughly be considered equivalent because we can always submit a new order and that order will immediately land at the end of the queue at the time of submission.

Including queue position information in the state space of reinforcement learning agent under market making setting is one main contribution of our work. Previous studies [3, 5, 8] claim that queue position information is essential for adverse selection risk control, but their main interest is in optimal execution of a single order under stochastic control framework. In addition, they only consider the queue position at best price level. However, we claim in Section 5 that queue position information at price levels beyond the best levels can have some additional values. Our main motivation to adopt model-free reinforcement learning technique is that analytically modeling the limit order book dynamics, inventory level, and multiple queue positions at the same time makes the problem extremely complex.

3.5 Reward Function

In this work, we investigated two different reward functions. The first is the one proposed by [12] where the reward given to the agent from time step t to $t+1$ is defined as

$$r_t = \phi_{bid,t} + \phi_{ask,t} + (1-\eta) \cdot I_t \cdot (p_{mid,t+1} - p_{mid,t}) \tag{4}$$

and $\phi_{bid,t}$ and $\phi_{ask,t}$ are defined as

$$\begin{aligned} \phi_{bid,t} &= (p_{mid,t+1} - p_{bid}) \cdot \mathbf{1}_{bid,t} \\ \phi_{ask,t} &= (p_{ask} - p_{mid,t+1}) \cdot \mathbf{1}_{ask,t} \end{aligned} \tag{5}$$

and I_t denotes inventory of the agent at time step t, $p_{mid,t}$ is the mid-price at time step t, $\mathbf{1}_{bid,t}$ ($\mathbf{1}_{ask,t}$) is an indicator functions of whether bid (ask) order is executed during time period from t to $t+1$, and p_{bid} (p_{ask}) is the price of bid (ask) order. The first two terms represent profit and loss from capturing bid-ask spread, and the last term represents profit and loss from price movements while holding inventory. This source of P&L is penalized by η ranging from zero to one to make the agent focus more on making profit from the first two terms and consequently learn to make profit from bid-ask spread. It is worth noting that if the bid (ask) limit order is executed and mid price does not change until the next time step $t+1$ or move up (down) , $\phi_{bid,t}$ ($\phi_{ask,t}$) would be positive (it is clear that $p_{bid} \leq p_{mid,t} \leq p_{ask}$), and negative if the mid price moves down (up). Based on our definition of adverse selection, we call the first case *normal fill*, and latter case *adverse fill*. Learning to make profit from the first two terms is equivalent to learning to control adverse selection risk.

The second reward function we investigated is from [4] where the reward from time step t to $t+1$ is defined as

$$r_t = \phi_{bid,t} + \phi_{ask,t} - \lambda \cdot |I_t|. \tag{6}$$

This reward function directly penalizes holding inventory. Deciding which value to use for the parameter λ can be tricky. If its value is too large, the agent would fear to hold even a single inventory and learn to always stay away from the best bid/ask level, leading to zero executions. If its value is negligible, the agent would ignore about controlling inventory risk and focus only on controlling adverse selection risk, which is harmful for final profit and loss. From our experience, it should be adequately chosen such that the positive return obtainable from the first two terms are about the same or slightly larger than the penalty from the last term. In our case the best performing parameter value was one hundredth of one tick size.

From our experiments, we conclude that using the second reward function leads to more stable inventory control, thus the experiment results presented in Section 5 are obtained under this reward function.

Table 1: Statistics of various order types

	Number	Proportion (%)
All Orders	562081 ± 188731	100 ± 0.00
Events	401597 ± 134201	71.52 ± 2.08
Market Orders	25066 ± 7385	4.52 ± 0.50
Limit Orders	295791 ± 95715	52.77 ± 0.82
Limit Orders at Best	207688 ± 65921	37.13 ± 1.44
Limit Orders at Non-Best	88102 ± 30943	15.64 ± 1.04
Cancel Orders	241223 ± 86361	42.69 ± 1.20
Cancel Orders at Best	168841 ± 61714	29.85 ± 1.36
Cancel Orders at Non-Best	72382 ± 25928	12.83 ± 1.21

3.6 Reinforcement Learning Algorithm

We have tested both rainbow Deep Q Network [7], and Proximal Policy Optimization (PPO) [11], and concluded that PPO performs better in terms of both stability and achieved total return. Thus the experiment results presented in Section 5 are obtained using PPO algorithm.

4 PRELIMINARY ANALYSIS

We use high-frequency limit order book data of KOSPI200 Index Futures starting from November 1, 2019 to January 31, 2020 which is comprised of 61 trading days. Each row contains information of the price/volume at each price level up to 5th level. Since the market dynamics can be significantly different at time close to the market open and end, we get rid of the first and last 15 minutes. For the training of deep reinforcement learning market making agent, we use the first 39 days for training and the last 22 days for testing. We do not use validation for reinforcement learning, but just train until moving average of episode returns stops increasing. Table 1 describes daily mean and standard deviation of various order types submitted during the full period.

4.1 Queue Position Analysis

In the Introduction, we have argued that order stacking strategy helps to reserve good queue positions. In this subsection, we quantitatively analyze the extent to which order stacking helps achieve good queue positions. For this purpose, we utilize a zero-intelligence agent who always stays at both best bid/ask price level, i.e., the agent always submits new order when the best bid/ask queue is *empty* and never cancels that order. The rationale behind using this zero-intelligence agent is to identify the marginal difference coming from order stacking framework, and not from other factors such as order placement behaviors of trained agents. We run 10 round-trip simulations over the entire data period and at each time step during the simulation, we collect the queue position snapshots at the best bid/ask price level. Figure 3 shows the best bid/ask queue position distributions under order stacking (blue) and no order stacking (orange) framework. We can clearly see that the density at queue positions close to zero is higher under order stacking, while no order stacking shows higher density at queue positions close to one.

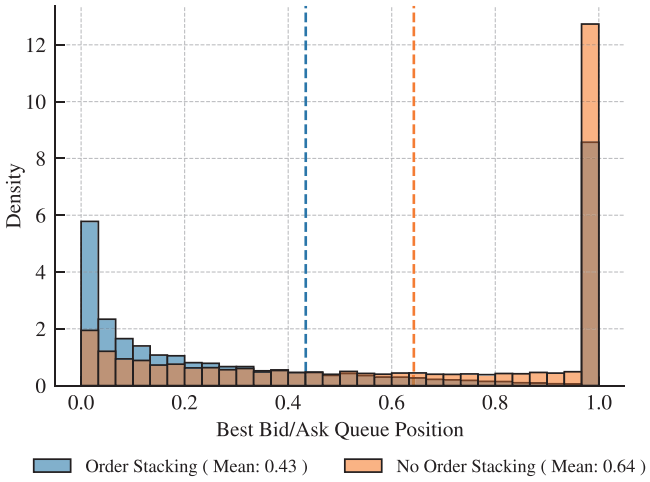

Figure 3: Queue position distributions at the best bid/ask price level under order stacking and no order stacking frameworks.

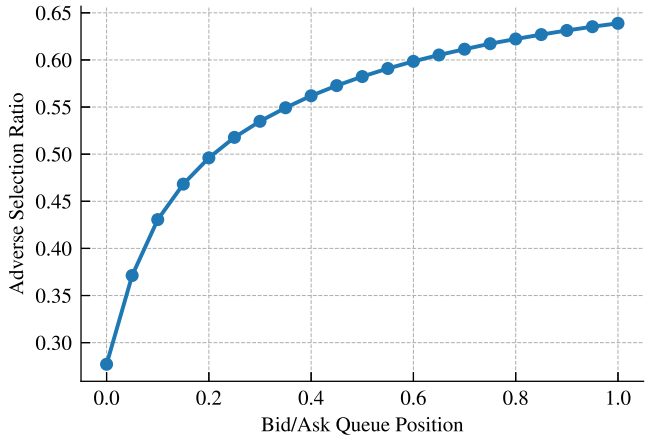

Figure 4: Ratio of adverse fills out of all fills conditioned on queue position.

4.2 Adverse Selection Risk Analysis

In this subsection, we use simulation approach to see the degree of adverse selection risk at random market state conditioned on queue position. This simulation approach is inspired from the work of [3]. The simulation details are as follow. At each time step, i.e. every 50 events, we artificially distribute our limit orders at various queue positions inside the queue at best bid/ask level. To be specific, at each time step we insert 21 limit orders each at best bid/ask level, of which the initial queue positions are $RQP \in \{0.05 \cdot i\}_{i=0,\ldots,20}$. Simultaneously, we check the status of the limit orders submitted at the previous time step, and for the orders that have been executed during the previous time period we label them as normal fills (executed orders that are not adverse fills) or adverse fills. Finally, we compute the ratio of adverse fills out of all fills. The adverse fill ratio is shown in Figure 4.

Table 2: Market making agents description

Models	Order Stacking	DRL	Full QPV
SRB_{NoOS}	X	X	X
SRB_{OS}	O	X	X
DRL_{NoOS} (1)	X	O	X
DRL_{OS} (1)	O	O	X
DRL_{OS} (5)	O	O	O

We point out two things from Figure 4. First, adverse selection risk is indeed high. At random market state, orders with queue positions larger than 0.2 have higher chance of being adverse filled than not. Second, queue position is indeed important to avoid adverse selection. We can see that the probability of orders at the end of the queue being adverse filled is higher than 0.6, while that of orders at the front of the queue is less than 0.3.

5 RESULTS

In this section, we present and compare the performances of various market making agents. We evaluate two Simple Rule-Based (SRB) agents and three Deep Reinforcement Learning (DRL) agents. The action behavior of the SRB agents is as follows: when the agent has no position, always place orders at both best bid/ask; when the agent has long position, always place order at best ask only; when the agent has short position, always place order at best bid only. With this simple logic, SRB agents are able to control inventory risk to some extent. We have also tested the zero-intelligence agent described in Section 4.1 who always places orders at both best bid/ask, but its performance is too poor even compared to the SRB agents. Thus, we decide not to include it here. In other words, even with the simple inventory control logic described above, the market making performance is significantly improved compared to zero-intelligence agent. The details of the five different market making agents are as follow, and summarized in Table 2.

- **SRB_{NoOS}**: The agent uses the simple inventory control logic, but does not utilize order stacking framework.
- **SRB_{OS}**: The agent uses the simple inventory control logic, and utilizes order stacking framework.
- **DRL_{NoOS} (1)**: The agent is trained under DRL settings described in Section 3, but does not utilize order stacking framework. Also, it only includes part of *QPV* that corresponds to the best bid/ask level in its state space.
- **DRL_{OS} (1)**: The agent is equivalent to **DRL_{NoOS} (1)** except that it utilizes order stacking framework.
- **DRL_{OS} (5)**: The agent is trained under DRL settings described in Section 3, and includes full *QPV* that corresponds to the 5 different price levels of bid and ask side. Also, it utilizes order stacking framework.

First, we analyze the inventory control behavior of the three DRL agents. We do not include the SRB agents here because the inventory control logic is explicitly defined with simple rules. Figure 5 shows the frequency of various inventory levels for the three DRL agents during the evaluation period. We conclude that all three agents are successful in learning proper inventory control policy. Frequencies of zero inventory are highest for all three agents,

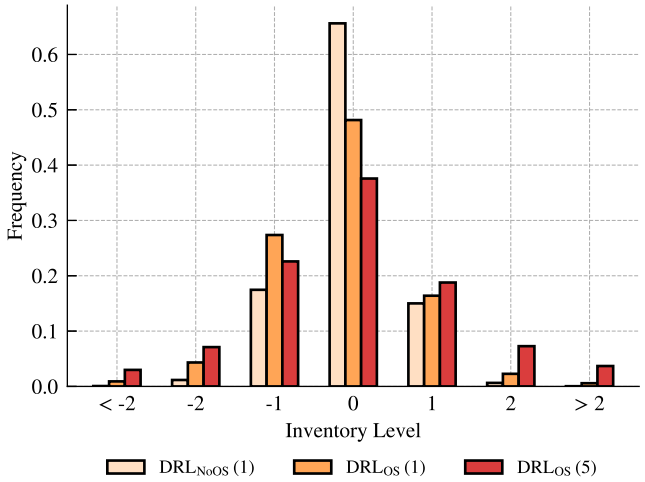

Figure 5: Frequency of various inventory levels for three different DRL market making agents.

Table 3: Trading statistics of each agent (daily average)

Models	P&L	P&L from Spread	Adverse Fill Ratio	Number of Fills
SRB_{NoOS}	-2.13	-8.49	0.551	3619
SRB_{OS}	2.33	-3.55	0.527	3716
DRL_{NoOS} (1)	3.15	1.69	0.487	1544
DRL_{OS} (1)	4.58	2.68	0.481	1929
DRL_{OS} (5)	5.57	3.30	0.483	2556

and inventory level rarely goes beyond 2 or -2. The frequencies of inventory level are approximately symmetric with respect to zero, meaning that the inventory level is overall mean-reverting. We can note that the frequency of zero inventory level is highest for DRL_{NoOS} (1), followed by DRL_{OS} (1) and DRL_{OS} (5). However, having lower frequency of zero inventory level does not necessarily mean that inventory risk is not properly controlled, but rather it can be interpreted as having more frequent executions. We can check from Figure 6 that indeed DRL_{OS} (5) has the highest number of daily executions, followed by DRL_{OS} (1) and DRL_{NoOS} (1).

Figure 6 presents evaluation results of the five different market making agents in various perspectives. The x-axis represents evaluation period of 21 dates. Blue lines indicate SRB agents. Orange lines indicate DRL agents. Dotted lines are used for agents without order stacking, while agents with order stacking are indicated by solid lines. To emphasize DRL_{OS} (5) and differentiate from others, red line is used for this agent. The upper left, upper right, lower left, and lower right subfigures show cumulative P&L, cumulative P&L from bid-ask spread, adverse selection ratio, and number of daily fills, respectively. It is worth reminding that the three terms in equation (4) with $\eta = 0$ sum up to the total P&L, while the first two terms represent P&L from capturing bid-ask spread. Higher P&L is obviously desirable, but analyzing P&L from bid-ask spread

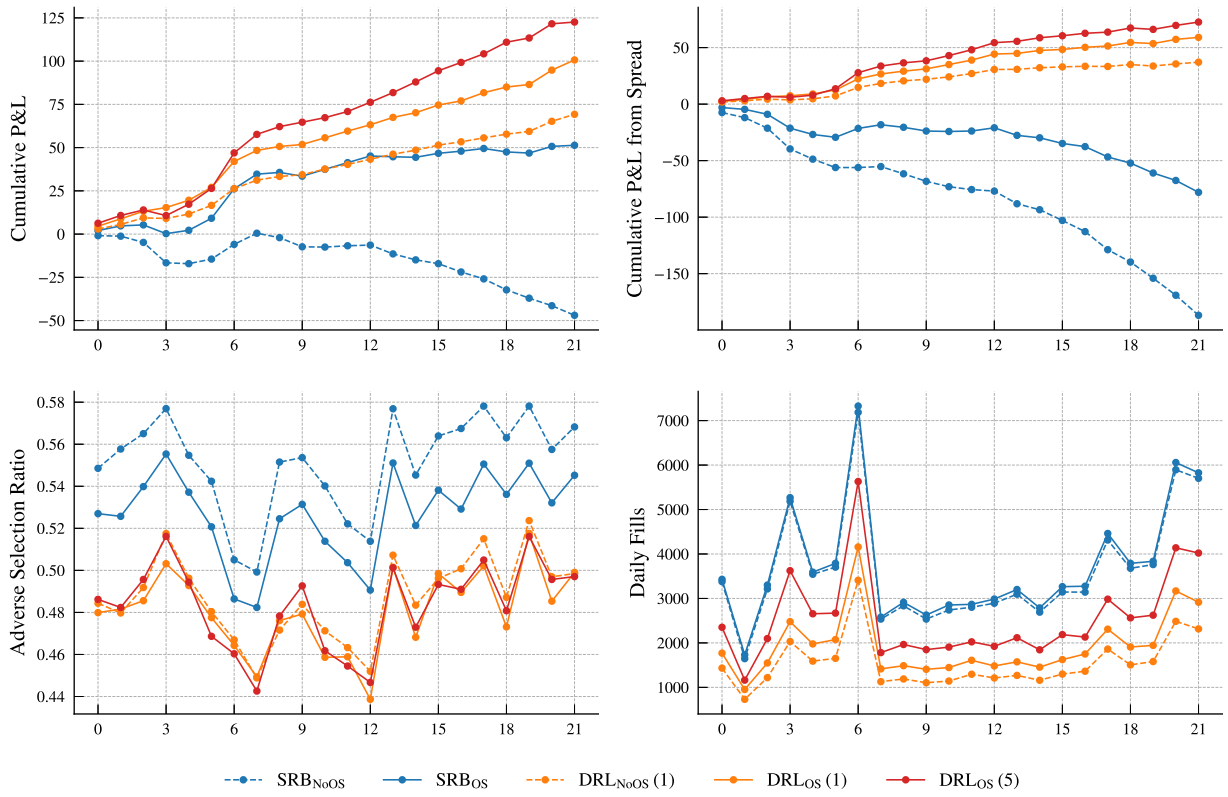

Figure 6: Evaluation of different market making agents in various perspectives. x-axis represents evaluation period of 21 dates.

separately is also important from market making perspective because the last term in equation (4) might include P&L coming from directional bets.

The results indicate that all three DRL agents achieve significantly lower adverse selection ratio compared to SRB agents, and they are all on average lower than 50%, of which the number can be checked in Table 3. Among them, DRL$_{OS}$ (5) agent achieves the highest performance in terms of both cumulative P&L and cumulative P&L from bid-ask spread, followed by DRL$_{OS}$ (1) and DRL$_{NoOS}$ (1). The difference in performance mainly comes from the number of daily fills. While maintaining about the same level of adverse selection ratio, DRL$_{OS}$ (5) is able to find more execution opportunities, presumably by utilizing the queue position information beyond the best price level. DRL$_{OS}$ (1) makes more fills compared to DRL$_{NoOS}$ (1) thanks to the help of order stacking. There would be not much difference between the action policies learned by DRL$_{OS}$ (1) and DRL$_{NoOS}$ (1) since they share the same state space and other details of reinforcement learning algorithm. However, better queue positions provided to DRL$_{OS}$ (1) naturally lead to more execution opportunities.

There is no significant difference between SRB$_{OS}$ and DRL$_{NoOS}$ (1) from the perspective of cumulative P&L, but DRL$_{NoOS}$ (1) shows significantly higher cumulative P&L from capturing bid-ask spread. This difference is coming from the difference in adverse selection ratio. Recall from Section 3.5 that maximizing P&L from capturing bid-ask spread is equivalent to minimizing adverse selection risk. This result is satisfactory because DRL$_{NoOS}$ (1) achieved significantly lower adverse selection ratio even without the help of order stacking. Thus, here we can conclude that DRL$_{NoOS}$ (1), with other two DRL agents, successfully learned not only to properly control inventory risk but also to control adverse selection risk.

We can see the pure benefits coming from order stacking by comparing SRB$_{NoOS}$ and SRB$_{OS}$ since these two agents utilize exact same action policy. The performance gap stems from the benefits of order stacking. While there is no significant difference in the number of daily fills, SRB$_{OS}$ outperforms SRB$_{NoOS}$ in all other three aspects. By utilizing order stacking framework, SRB$_{OS}$ is provided with better queue positions on average, leading to lower adverse selection ratio. Eventually, SRB$_{OS}$ is able to achieve higher P&L and higher P&L from bid-ask spread, even with the same inventory control policy.

We conclude that order stacking provides better queue positions and this can help lower adverse selection ratio. However, it may not be sufficient without proper adverse selection risk control that takes into account the queue position information of resting limit orders. With proper control of adverse selection risk, order stacking can provide more opportunities of executions which can possibly lead to better performance in terms of both P&L and P&L from bid-ask spread, while still maintaining the adverse selection ratio low.

6 CONCLUSION

Optimal market making strategy should be able to simultaneously control inventory risk, adverse selection risk, and non-execution risk. For better control of the latter two risks, reserving good queue positions is important. For this purpose with focus on better control of adverse selection risk, we adopt order stacking framework which is to place limit orders at each price level beyond the best bid/ask price. While this framework guarantees better queue positions, to fully exploit this framework we should be able to consider queue position information of stacked orders. In this work we formalize the order stacking framework and develop a *Queue Position Vector, QPV* that efficiently encodes information of multiple queue positions. We show that a reinforcement learning approach with *QPV* in its state space leads to better control of adverse selection risk, and supporting it with order stacking further improves the performance of market making.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1A2C1010677 and No. NRF-2019R1C1C1010456).

REFERENCES

[1] Marco Avellaneda and Sasha Stoikov. 2008. High-frequency trading in a limit order book. *Quantitative Finance* 8, 3 (2008), 217–224.
[2] Alvaro Cartea, Ryan Donnelly, and Sebastian Jaimungal. 2018. Enhancing trading strategies with order book signals. *Applied Mathematical Finance* 25, 1 (2018), 1–35.
[3] Xuefeng Gao and Tianrun Xu. 2022. Order scoring, bandit learning and order cancellations. *Journal of Economic Dynamics and Control* 134 (2022), 104287.
[4] Bruno Gašperov and Zvonko Kostanjčar. 2021. Market making with signals through deep reinforcement learning. *IEEE Access* 9 (2021), 61611–61622.
[5] Federico Gonzalez and Mark Schervish. 2017. Instantaneous order impact and high-frequency strategy optimization in limit order books. *Market Microstructure and Liquidity* 3, 02 (2017), 1850001.
[6] Martin D Gould and Julius Bonart. 2016. Queue imbalance as a one-tick-ahead price predictor in a limit order book. *Market Microstructure and Liquidity* 2, 02 (2016), 1650006.
[7] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*.
[8] Charles-Albert Lehalle and Othmane Mounjid. 2017. Limit order strategic placement with adverse selection risk and the role of latency. *Market Microstructure and Liquidity* 3, 01 (2017), 1750009.
[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
[10] Ciamac C Moallemi and Kai Yuan. 2016. A model for queue position valuation in a limit order book. *Columbia Business School Research Paper* 17-70 (2016).
[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
[12] Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. 2018. Market making via reinforcement learning. *arXiv preprint arXiv:1804.04216* (2018).
[13] Chaiyakorn Yingsaeree. 2012. *Algorithmic trading: Model of execution probability and order placement strategy*. Ph. D. Dissertation. UCL (University College London).
[14] Muchen Zhao and Vadim Linetsky. 2021. High frequency automated market making algorithms with adverse selection risk control via reinforcement learning. In *Proceedings of the Second ACM International Conference on AI in Finance*. 1–9.
[15] Yueyang Zhong, Yee Man Bergstrom, and Amy Ward. 2021. Data-driven market-making via model-free learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 4461–4468.