Deep Recurrent Q-Networks for Market Making

Pankaj Kumar

Copenhagen Business School, Denmark
pk.mpp@cbs.dk

Abstract. Market Making is high frequency trading strategy in which an agent provides liquidity simultaneously quoting a bid price and an ask price on an asset. Market Makers reaps profits in the form of the spread between the quoted price placed on the buy and sell prices. Due to complexity in inventory risk, counterparties to trades and information asymmetry, understating of market making algorithms is relatively unexplored by academicians across disciple. In this paper, we develop realistic simulations of limit order markets and use it to design a market making agent using Deep Recurrent Q-Networks. Our approach outperforms a prominent benchmark strategy from literature, which uses temporal-difference reinforcement learning to design market maker agents. The agents successfully reproduce stylized facts in historical trade data from each simulation.

Keywords: Deep Reinforcement Learning; Market Making; Limit Order Books; High Frequecy Tading Stratergies; Agent Based Models.

1 Introduction

The electronification of securities trading has transformed traditional human-driven markets into predominantly automated, where high frequency trading (HFT) typically exceeds 80% of total volume traded in U.S listed equities [16, 17]. HFT is a form of automated trading in which security positions are turned over very quickly by leveraging advanced technology and the associated extremely low latency rates [18]. Market Making is HFT based strategies contributing to market liquidity by matching buyer and seller orders. The profit is earned as the spread between the quoted price placed on the buy and sell prices. With every-growing minuscule limit order book (LOB) data, complexity in inventory risk, counterparties to trades and information asymmetry, understating of market making algorithms is relatively shallow [3, 2, 26]. This paper uses a variant of Deep Recurrent Q-Networks (DRQN) to design market making agents interacting with realistic limit order book simulation framework.

A number of market making strategies have been proposed across disciple, including finance [3, 7], econophyics [16] and machine learning[5, 2, 26]. Earlier work in finance considers maker making as a problem of stochastic optimal control, where order book dynamics are designed using control algorithms after developing the arrival and execution model [3, 6] to understand the price impact, adverse selection, risk measures, and inventory constraints.

Another prominent approach, agent based model (ABM), ranging from zero intelligence to intelligent variants are used to study market making, but are typically evaluated in simulated markets without using real market data. It gives the modeler flexibility to churn out potentially emergent phenomenon as a result of interaction between agents. With evolving technology-based disruption in HFT, the existing learning models and empirical models are deficient and may no longer be appropriate. Reinforcement learning (RL) has been applied for market making [26], algorithmic trading [31], optimal execution [20], and foreign exchange trading [9]. However, defining hand-crafting features in reinforcement learning for agents to learn while interacting within a dynamic environment is a major throttle block. Also, RL could be slow to learn in large state spaces and the methods did not generalize (across the state space).

Deep learning eliminates the need for manual feature design, thus finding compact representations in high-dimensional data. It also helps to generalize across states improving the sample efficiency for large state-space RL problems. Augmenting deep learning with reinforcement learning, deep reinforcement learning (DRL), enables RL to scale to problems with high-dimensional state and action spaces. The outstanding success stories of DeepMind's, kick-starting with superhuman level performance in Atari 2600 video games [19], AlphaGo [25], and AlphaStar [28] proves the effectiveness of DRL. However, only a few works is featured optimal execution [21], market making [10], and high frequency trading [31] as compared to the games.

The success of such single DRL's can be accredited to the use of experience replay memories, which legitimate Deep Q-Networks (DQNs) to be trained efficiently through sampling stored state transitions. However, despite the ever-increasing performance on popular benchmarks such as Atari 2600 games, DQN struggle to generalize when evaluated in different environments. It does not perform well in partially observable domains [13], overestimate action values under certain conditions [12], and not efficient when experience replay needs to be prioritized [24]. Deep Recurrent Q-Networks (DRQN) [13] proposed using recurrent neural networks, in particular, LSTMs (Long Short-Term Memory) solves the above problem by replacing the first post-convolutional fully connected layer with an LSTM layer in DQN setting. With this incorporation, DRQN has memory capacity so that it can even work with only one input rather than a stacked input of consecutive frames. Double DQN [12] obliterate the overestimation problem in DQN, resulting in more stable and reliable learning. By prioritizing experience, authors [24] achieved a new state of art human-level performance across benchmark Atari games.

The main contribution of this paper is to develop realistic simulations of limit order markets and use it to design a market making agent using DRQN. The simulation framework takes account of the agent's latency and have build-up maker/taker fees as defined in NYSE. We modify the classical DQRN architecture and incorporate double Q-learning and prioritized experience to take account of volatile, illiquid and stagnant markets. Our approach outperforms a prominent benchmark strategy from literature, which uses temporal-difference reinforcement learning to design market maker agents.

2 DRL SIMULATION FRAMEWORK

2.1 Environment: Simulation Framework.

We have designed a simulation framework over realistic market design, market engine, communication interface, and the Financial Information eXchange (FIX) protocol [1], an open standard that is used extensively by global financial markets. This framework is unconstrained on historical data, represents realistic exchange and makes no assumptions about the market. We have designed a simulation framework over realistic market design, market engine, communication interface, and the Financial Information eXchange (FIX) protocol [2], an open standard that is used extensively by global financial markets. This framework is unconstrained on historical data, represents realistic exchange and makes no assumptions about the market. From a high-level perspective, the simulation framework comprises of two entities, agents and market, as shown in Figure 1.

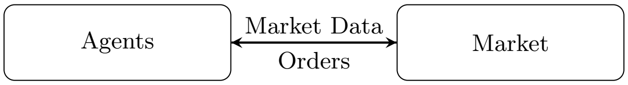


Fig. 1: High-level simulation framework.

Markets act as communication nodes, which listen for agents to make connections and process incoming orders, aggregating to order books, creating trades according to a matching engine designed for each instrument, etc. Matching engines provide the transactional integrity for an electronic trading venue, marketplace, or exchange using various algorithms to facilitate the matching of buyers and sellers. The most common is price/time priority or First In First Out (FIFO). FIFO ensures that all orders at the same price level are filled according to time priority.

2.2 Agents: Trading Strategies.

In our simulation framework, we populate the market with two types of agents, namely, *market makers* and *market takers*. The agents interact with the market using order type, price, and quantity according to their internal logic. The submitted order in limit order book is then matched using price-time priority

[1] FIX Trading Community, ”Financial Information eXchange (FIX) Protocol,” https://www.fixtrading.org/.

[2] FIX Trading Community, ”Financial Information eXchange (FIX) Protocol,” https://www.fixtrading.org/.

algorithms from matching engine. The latency manager build in the simulation framework manages the whole history with suitable timestamp for order to help the agents maintain their inventory tight. The trading strategies of the two agents are discussed below.

Maker's Strategy: In this paper, we implement a realistic market making strategy taking account of the order size which was missing in the past literature [26]. It is roughly based on the liquidity providing strategy described in a prominent research study [14].

At each event time t, the total quantity of liquidity Q_t market maker willing to provide a fixed proportion of their available capital C_t is defined as:

$$Q_t = \omega C_t \tag{1}$$

The market maker's available capital, C_t, comprises of starting capital plus the profits accumulated from buy and sell trades up to time t, and the profit or loss from the remaining inventory holdings:

$$\begin{aligned} C_t = C_0 & \\ &+ \min\left(\sum_{m,t-1} d_i^b - \sum_{m,t-1} d_i^a\right)\left(\frac{\sum_{m,t-1} d_i^a p_i^a}{\sum_{m,t-1} d_i^a} - \frac{\sum_{m,t-1} d_i^b p_i^b}{\sum_{m,t-1} d_i^b}\right) \\ &+ \left[\max\left(\sum_{m,t-1} d_i^b - \sum_{m,t-1} d_i^a\right) - \min\left(\sum_{m,t-1} d_i^b - \sum_{m,t-1} d_i^a\right)\right]\bar{p}_{t-1} \end{aligned} \tag{2}$$

where, m_t is the history of all trades in time t, d_t is liquidity demand in time t, $p_t^{a,b}$ is bid/ask price of a asset at time t and $\bar{p}_t$ is observed market mid-price at time t.

The limit order size that a market maker is willing to buy or sell is defined as:

$$q_t^{a,b} = \left(\frac{1}{2}Q_t\right) \cdot \left(\frac{1}{N}\right) \cdot \left(\frac{I \pm \bar{I}}{\bar{I}}\right) \cdot \varUpsilon\left(1 - \left|\frac{\bar{p}_{t-1}}{\bar{p}_{t-t_c}}\right|\right) \tag{3}$$

where, I_t, $\bar{I}$ are inventory at time t and maximum inventory respectively; $\varUpsilon$ for accounts for adverse selection and t_c is price change period.

The total quantity of liquidity Q_t market maker wish to supply is equitably chopped into buy and sell orders. This is additionally subdivided into N distinct limit orders on each side of the order book. The splitting is done to adjust inventory at an optimum level. The risk of adverse selection is accounted for using the last term with $\varUpsilon$ as a parameter. For detail discussion on the variables defined above is carried on in the research article [14].

At each event time, t, market maker update parameters, $\varTheta_t^a$ and $\varTheta_t^b$, which is required for deriving relative prices, $\mathbb{D}_t^{a,b}$. The market makers encounters adverse selection risk when posting limit orders at fixed distance from mid price $\bar{p}_t$. The volatility of mid prices in the previous five periods is chosen as proxy for the risk with Λ governing the sensitivity of the bid-ask spread to volatility. The bid-ask spread is set as linear function of above volatility subject to minimum constant, ξ. After setting bid/ask, further orders are placed either side of book above/below former price using parameter τ. The pricing strategy for ith limit order of market maker is defined by equations below:

$$p_t^{a,b,i} = \bar{p}_t + \Theta_t^{a,b} \cdot \mathbb{S}_t \pm \min\Big(\Lambda\sigma_{(t-1:t-5)}, \xi\Big) \pm i \cdot (10^{-\tau}), \tag{4}$$

where, spread, $\mathbb{S}_t$, is a moving average of the market half-spread [26].

The market making agents at each event time t can clear its outstanding orders in the limit order book using two criteria. The first corresponds to clearing its inventory using a market order if it has not been executed at the end of trading. The second criterion, instead, takes account of the current market condition to remove the order from the limit order book. In particular, we define a cancellation probability as:

$$\Psi_t = 1 - \exp^{-\psi \cdot vol_p} \tag{5}$$

where, ψ is sensitivity parameter and vol_p is the perceived volatility [4].

The market maker follows simple heuristics to cancel orders in the limit order book. First, the range of ∓20 from the most recent transaction price is identified. Then, the existing order is investigated for outside price range. If there are orders outside the range, then the Ψ_t percent of the order will be canceled from the order book, rest remaining.

Taker's Strategy: As discussed above, the market takers wishes to fill his/her trade immediately by agreeing with the currently listed prices on the order book. While trading large asset over the day, the market taker tends to minimize price impact and trading cost. In our simulation framework, market makers follow momentum strategy. A momentum strategy is modern equivalent to classical day traders, who earn profits from market movements by taking liquidity aggressively. In this simple momentum trading strategy, the trend is captured using a price change rate, $\Delta p_t = \frac{p_{(t)} - p_{(t-t_c)}}{p_{(t-t_c)}}$, where, p_t is the price of the asset at time t and t_c is the price change period.

The size of the market order is proportional to the strength of the price rate change subject to inventory constraint. That is, the size of the market order will be:

$$d_{MK,t} = (\delta) \cdot (\Delta p_t) \cdot \left(1 - \left(\frac{I_{MT,t-1}}{\bar{I}_{MT}}\right)^h\right) \tag{6}$$

where δ is sensitivity of order size to price movement parameter, $I_{MT,t}$ is market taker's inventory at time t, $\bar{I}$ is maximum inventory and h controls the order size as as $I_{MT,t}$ approaches to $\bar{I}$.

2.3 State Representation.

The state representation is composed of *agent-state* and *market-state*, such that it contained the information about agent's position as well as market features. The agent-state is described by following variables: Inventory at time t, I_t; Active quoting distances, normalized by spread, $\mathbb{S}_t$; Market maker's update parameters, Θ^a_t and Θ^b_t, which is required for deriving relative prices, $\mathbb{D}^{a,b}_t$; Past price history, n_h, to recognize the market trend or risk; and Cancellations probability, Ψ_t, used to clear outstanding orders in the limit order book.

The complexity of the market is represented by the market state, which contains the partial observable state of the limit order book at each event period, as well as any prior information from previous periods. In this paper, we include the following market features, as which are described in benchmark paper [26]: Bid-Ask spread; Mid-price move; Queue imbalance [6]; Volume imbalance [30]; Orderbook depth[30]; Signed volume; Perceived volatility [4] and Relative strength index.

2.4 Action Space.

In our market making setting, the possible actions the agent has to decide consists of four options, "buy," "hold," "sell," and "cancel". The agent can buy/sell fixed multiple of integer values at particular price $p^{a,b}_t$ at time t. The cancellation also can be done in only integer values. At the end of trading, the market maker clears its inventory using a market order if it's not executed or canceled.

2.5 Reward Functions.

The reward function in this paper is traditional profit and loss (PnL), which keeps track of money gained or lost. The agents try to maximize the profits it accumulates during a trading day subjects to the inventory. To incorporate realism as per the existing market design, the marker-taker fee model is also included. The market-taker fee model is a pricing structure in which an exchange customarily pays its members a per-share rebate to supply (i.e., "make") liquidity and levy on them a fee to remove (i.e., "take") liquidity. For example, the agents may be charged 0.0030 per share for taking liquidity from the market (i.e., 3 dollars per 1000 shares) and gets a rebate of 0.0020 per share for posting liquidity (i.e., 2 dollars per 1000 shares). For more details on fee structure, please refer to the official website of NYSE [3].

At a given event time t, lets us assume that market making agents post buy/ sell limit order of size $q^{a,b}_t$, he/she receives execution confirmation at time $t+\Delta t$. The Δt is vaguely referred as latency provided by the simulation framework, as there is always time a lag between a request made and actual transaction done. The much ignored transaction costs in academic literature is incorporated using

[3] https://www.nyse.com/markets/nyse/trading-info/fees

an exponential penalty on the number of shares executed and maker/taker fee as defined in NYSE. Notably, we define the PnL function as:

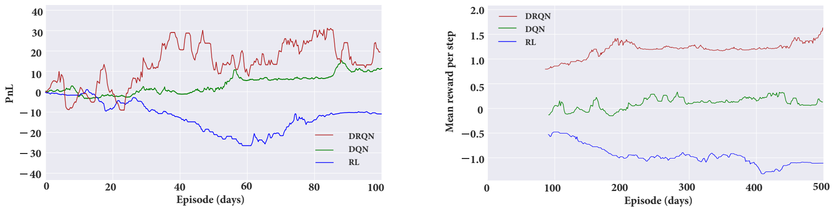
$$\mathbb{R}_{PnL}(t) = \sum_{m_t} \left(q_t^a p_t^a - q_t^b p_t^b\right) - \alpha \left(\exp^{\frac{q_t^{a,b}}{\Delta t}}\right) \pm \beta \left(F_{(maker/takerfee)}\right) \tag{7}$$

3 Experiments and Results

We run the model for 1000 iterations to find relevant hyper-parameter using random search. After that, we train the models for some ten million time steps for intervals of 10000, which is equivalent to 500 trading days to collect data, monitor and visualize the learning of the agent. Then, testing the environment on the benchmark to see the agent's learning pattern.

3.1 Results and Analysis.

The performance of the agents is compared in Figure 2. In-spite of handcrafted strategy, where actions with various quantities are taken at different states, the RL agent performs badly and not stable as compared to DRQN and DQN agents. It is to be noticed that the trading strategy which RL agents follow doesn't take account of order size, cancellation, adverse selection, transaction cost and volatility, which the current simulator introduces while interaction. Adding to the same, the order matching is subject to market-takers, who trades on market trends as described in agent's trading strategies. DQN performance is stable , but fails to outperform the DRQN. The reasoning may be linked to not efficient state representation, overestimated action values, partial observability and pritorized experience, which DRQN incorporates. The same is reflected in Figure 2. To understand the performance better, we need to action selection with respect to limit order book dynamics, which we plan to do next.



(a) Reward.

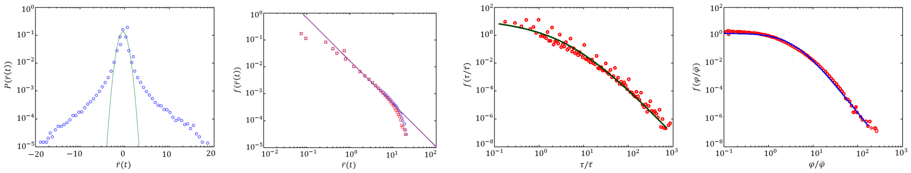
(b) Mean Reward.

Fig. 2: Trading agent performace.

3.2 Validation.

In agent-based models of financial markets, it is standard practice to measure the validity of the model by investigating whether the order-book data have particular characteristics, known as the "stylized facts" [1]. We present some of the stylized facts reproduced from historical trade data.

To reproduce stylized facts concerning price, we first calculate return, which is given by $r(t) = \log(p_t) - \log(p_t)$. The heavy tails (HT) in the distribution of returns is depicted in Figure 3a. The normalized return distribution has a fatter tail than green Gaussian distribution. Furthermore, the cumulative distributions function [8, 1], shown as the blue (positive tail) and red (negative tails) in Figure 3b, exhibits power law (PL). The violet line is the asymptotic power-law function with tail exponent 4.



(a) HT. (b) PL. (c) LOS. (d) LOC.

Fig. 3: Stylized facts.

We now switch from price to order size. The Figure 3c illustrates the probability density distribution (PDF) $f(\tau/\bar{\tau})$ of limit order size (LOS) τ, where $\bar{\tau}$ is mean order size of individual stock. The green line is Gamma distribution fit to the normalized order size. It is evident from the figure that the Gama distribution fits remarkably good to empirical PDF. This is in line with the existing literature [1], confirming the existence of heavy tail in limit order size. The limit order cancellation (LOS) also follows Gama distribution which can be seen in Figure 3d. The fitting procedure is the same as the limit order size.

4 Conclusions

In this paper, we have designed a market making agent using deep recurrent Q-network that outperforms a prominent benchmark strategy, which uses temporal-difference reinforcement learning. The market making agents interact with highly realistic simulation of the limit order book, which till now is non-existence in the academic research. The suitable modification in the exciting DRQN network architecture [13] and training procedure allowed our agents to yield predominant performance. It paved the way for researchers to include latency in the agent's strategy and extend to portfolio with suitable hedging strategies rather than single asset. Another direction is to incorporate order book data with deep reinforcement learning, and extend it to a multi-agent setting, where all agents learn and trade simultaneously.

References

1. Abergel, F., Anane, M., Chakraborti, A., Jedidi, A., Muni Toke, I.: Limit Order Books. Physics of Society: Econophysics and Sociophysics, Cambridge University Press (2016)
2. Abernethy, J., Kale, S.: Adaptive market making via online learning. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 2058–2066. Curran Associates, Inc. (2013)
3. Avellaneda, M., Stoikov, S.: High-frequency trading in a limit order book. Quantitative Finance **8**(3), 217–224 (2008)
4. Bartolozzi, M.: A multi agent model for the limit order book dynamics. The European Physical Journal B **78**(2), 265–273 (Nov 2010)
5. Brahma, A., Chakraborty, M., Das, S., Lavoie, A., Magdon-Ismail, M.: A bayesian market maker. In: Proceedings of the 13th ACM Conference on Electronic Commerce. pp. 215–232. EC '12 (2012)
6. Cartea, A., Jaimungal, S., Ricci, J.: Buy low sell high: A high frequency trading perspective. SIAM Journal on Financial Mathematics **5** (11 2011)
7. Chakraborty, T., Kearns, M.: Market making and mean reversion. In: Proceedings of the 12th ACM Conference on Electronic Commerce. pp. 307–314. EC '11 (2011)
8. Cont, R.: Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance **1**(2), 223–236 (2001)
9. Dempster, M., Leemans, V.: An automated fx trading system using adaptive reinforcement learning. Expert Systems with Applications **30**, 543–552 (04 2006)
10. Elwin, M.: Simulating market maker behavior using Deep Reinforcement Learning to understand market microstructure. Master's thesis, KTH Royal Institute of Technology, Stockholm (January 2019)
11. Gould, M.D., Porter, M.A., Williams, S., McDonald, M., Fenn, D.J., Howison, S.D.: Limit order books. Quantitative Finance **13**(11), 1709–1742 (2013)
12. Hasselt, H.v., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 2094–2100. AAAI'16, AAAI Press (2016)
13. Hausknecht, M., Stone, P.: Deep recurrent q-learning for partially observable mdps. In: AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15) (November 2015)
14. Karvik, G.A., Noss, J., Worlidge, J., Beale, D.: The deeds of speed: an agent-based model of market liquidity and flash episodes. Bank of England working papers 743, Bank of England (Jul 2018)
15. Li, Y.: Deep reinforcement learning. CoRR **abs/1810.06339** (2018)
16. McGroarty, F., Booth, A., Gerding, E., Chinthalapati, V.L.R.: High frequency trading strategies, market fragility and price spikes: an agent based model perspective. Annals of Operations Research (Aug 2018)
17. Menkveld, A.: High frequency trading and the new market makers. Journal of Financial Markets **16**(4), 712–740 (2013)
18. Menkveld, A.: The economics of high-frequency trading: Taking stock. Annual Review of Financial Economics **8**(1), 1–24 (2016)

19. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (Feb 2015)
20. Nevmyvaka, Y., Feng, Y., Kearns, M.: Reinforcement learning for optimized trade execution. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 673–680. ICML '06 (2006)
21. Ning, B., Ling, F.H.T., Jaimungal, S.: Double deep q-learning for optimal execution. ArXiv **abs/1812.06600** (2019)
22. OroojlooyJadid, A., Hajinezhad, D.: A review of cooperative multi-agent deep reinforcement learning (2019)
23. Penalva, J., Cartea, A., Jaimungal, s.: Algorithmic and High-Frequency Trading. Cambridge University Press, first edn. (08 2015)
24. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. In: International Conference on Learning Representations. Puerto Rico (2016)
25. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nature **529**(7587), 484–489 (Jan 2016)
26. Spooner, T., Fearnley, J., Savani, R., Koukorinis, A.: Market making via reinforcement learning. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. pp. 434–442. AAMAS '18, Richland, SC (2018)
27. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, second edn. (2018)
28. Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W.M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., Ewalds, T., Horgan, D., Kroiss, M., Danihelka, I., Agapiou, J., Oh, J., Dalibard, V., Choi, D., Sifre, L., Sulsky, Y., Vezhnevets, S., Molloy, J., Cai, T., Budden, D., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Pohlen, T., Wu, Y., Yogatama, D., Cohen, J., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Apps, C., Kavukcuoglu, K., Hassabis, D., Silver, D.: AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. `https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/` (2019)
29. Watkins, C.J., Dayan, P.: Technical note: Q-learning. Machine Learning **8**(3), 279–292 (May 1992)
30. Weber, P., Rosenow, B.: Order book approach to price impact. Quantitative Finance **5**(4), 357–364 (2005)
31. Wei, H., Wang, Y., Mangu, L., Decker, K.: Model-based reinforcement learning for predictions and control for limit order books. ArXiv **arXiv:1910.03743** (2019)