Towards a fully RL-based Market Simulator

Leo Ardon
J.P. Morgan AI Research
leo.ardon@jpmorgan.com

Nelson Vadori
J.P. Morgan AI Research
nelson.n.vadori@jpmorgan.com

Thomas Spooner
J.P. Morgan AI Research
thomas.spooner@jpmorgan.com

Mengda Xu
J.P. Morgan AI Research
mengda.xu@jpmorgan.com

Jared Vann
J.P. Morgan AI Research
jared.vann@jpmorgan.com

Sumitra Ganesh
J.P. Morgan AI Research
sumitra.ganesh@jpmorgan.com

ABSTRACT

We present a new financial framework where two families of RL-based agents representing the Liquidity Providers and Liquidity Takers learn simultaneously to satisfy their objective. Thanks to a parametrized reward formulation and the use of Deep RL, each group learns a shared policy able to generalize and interpolate over a wide range of behaviors. This is a step towards a fully RL-based market simulator replicating complex market conditions particularly suited to study the dynamics of the financial market under various scenarios.

CCS CONCEPTS

• **Theory of computation** → **Multi-agent reinforcement learning**; **Market equilibria**; • **General and reference** → *General conference proceedings.*

KEYWORDS

multi-agent, reinforcement learning, market making

ACM Reference Format:

Leo Ardon, Nelson Vadori, Thomas Spooner, Mengda Xu, Jared Vann, and Sumitra Ganesh. 2021. Towards a fully RL-based Market Simulator. In *2nd ACM International Conference on AI in Finance (ICAIF'21), November 3–5, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3490354.3494372

1 INTRODUCTION

Being able to understand the dynamics of the financial market has always been the Holy Grail of many economists and financial researchers. However, the number of actors at play and the variety of behaviors present in the market, make it very hard to fully understand the interplay between all the market participants. In the quest to solve the market making problem for instance, where the crux of the problem lies in understanding the interaction between the Liquidity Provider (LP), the Liquidity Taker (LT) and the Electronic Communication Network (ECN), two main approaches have emerged. The first one, more statistical, considers the different actors in isolation and make assumptions about the rest of the market. The other strategy is to model the market as a multi-agent system (MAS) where the participants are represented as independent entities able to interact between each other.

While this method seems more suited to help uncover phenomena emerging from the actions performed by multiple participants, modeling the agents' behavior is hard. Hand-coded policies driven by business experience and common sense are typically used but are either not complex enough to truly characterize the agent's behavior or are too difficult to calibrate with the data available.

With the recent breakthroughs in Reinforcement Learning (RL) and the development of RL-based multi agent systems, researchers have started to model the market participants as learning agents providing a less opinionated setting to study the dynamics of the market. By only specifying a well-designed reward function one can represent a wide range of behaviors without having to make hard assumptions on the policy to apply: the agent learns by itself what action to take in order to maximize its cumulative reward, based on the rules of the game only.

In this paper we build upon the work from [10] with their use of RL to model the Liquidity Provider (or market maker) and introduce a new financial framework principally composed of RL-based agents able to learn how to react to a change in the market. We propose a unified reward formulation for the two groups of RL-based agents, each of which having its own family of behaviors balancing the trade-off between quantity and PnL. We leverage the theoretical framework from [16] to train concurrently two shared policies for the two families of agents present in the system, namely the Liquidity Takers and the Liquidity Providers. *The parametrized formulation of the reward functions and the use of Deep Reinforcement Learning to train the shared policies, take full advantage of the generalization power of neural networks to learn over an entire spectrum of behaviors.*

1.1 Related Work

The work of Garman [11] was one of the first to present a stochastic model of the market making problem. He presented a framework where the market is centralized around one LP, who has the monopoly of all trading. He assumed that the aggregated supply and demand are exogenous to the LP and are Poisson distributed in time. This work was foundational to many subsequent research on the matter and paved the way to many discoveries. However, this work only considers a single LP and doesn't take into account the competition effect that multiple LPs have on each other. The fixed distribution of supply and demand does not truly reflect the reality of the financial market. In this configuration the LT are not able to react to the LP's actions. Building upon this theory, Amihud and

ICAIF'21, November 3–5, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-9148-1/21/11…$15.00
https://doi.org/10.1145/3490354.3494372

Mendelson [1] and later on Guéant, Lehalle and Fernandez-Tapia [12] have focused on the importance of inventory (and the risk associated with it) to find the optimal pricing for market making. These papers tackle the problem of the temporal discrepancy between the orders executions and the continuous price change leading to the price risk a LP bears by holding inventory, but also ignore the competition effect among the agents.

The use of multi-agents systems to model the market participants has started to emerge more recently. In [9], Das successfully replicates important features of real financial time series by simulating the markets in a multi-agents system. The traders agents, corresponding to the LTs, are divided into two groups: the "informed" traders who have an idea of the fundamental value of the asset and trade accordingly; and the "liquidity" traders sometimes called Zero Intelligence (ZI) agents who trade randomly. These simple models of traders were sufficient to study simple financial facts but bring some limitations to the analysis of more complex market properties. The work of Cui and Brabazon ([8]) and [6] highlighted the necessity of intelligent agents to replicate real market conditions. In [18] and [17] for instance, the authors successfully replicate some "stylized facts" observed in the markets by enhancing ZI agents with hand-coded heuristics. We argue that these heuristics only cover commonly known behaviors but do not capture more complex and undisclosed market strategies.

RL-based frameworks to solve the market making problem have been proposed. [5] and more recently [14] and [10] used reinforcement learning to train financial agent in a simulated market making environment. These approaches considered a single type of RL-based learning agent (the ECN or the LP) simplifying the model of the other market participants to heuristic based hand-coded policies. In our approach we relax the modeling assumptions made on the LT agent type to let emerge more natural behaviors.

1.2 Our Contributions

In this paper, **(i)** we formalize two types of market participants as RL-based agents, able to adopt different behaviors via a parametrized family of reward functions (sections 3.3, 3.4). **(ii)** We train simultaneously two groups of RL based financial agents, each using a shared policy (section 3.2) capable of interpolating over a range of behaviors. **(iii)** Finally, we perform an extensive study of the impact that the RL-based Liquidity Takers have on the pricing strategy of the Liquidity Providers (section 4). In particular, we show that the Liquidity Provider agent learns to adapt his strategy depending on the type of investor he is connected to while the Liquidity Takers agent learns to react to different pricing strategy.

2 BACKGROUND

We focus our research to the context of an order-driven market, where a single security is being traded for cash. Although many actors take part in this market, we can distinguish three types of participant: the Liquidity Provider (LP) also known as market maker, the Liquidity Taker (LT) or investor, and the Electronic Communication Network (ECN). The role of the Liquidity Provider is to ensure fluidity in the market by offering liquidity to buyers and sellers. LPs continuously quote *bid* (buy) and *ask* (sell) prices at which they are willing to trade. The LPs adapt their quotes dynamically to limit their exposure to the price variations caused by the fluctuation of supply and demand. Finding the optimal pricing strategy is refer to as *market making*. The LP is connected to both LTs and ECNs and can trade directly with both of them. The LTs are the consumers; they enter the market to execute orders. They received the prices streamed by the LPs and ECNs and decide with whom they wish to trade. They typically trade directly with the LP because it offers better prices than going via the ECN. The ECN plays an important role in the financial market by centralizing buy and sell orders. It can be seen as a collection of FIFO[1] queues of *orders* placed at different levels by the market participants and waiting to be fulfilled. A level corresponds to the price at which the order will be executed. In general the ECN gives a good sense of how the market is evolving. It is often used as a proxy to represent all the other market participants.

2.1 Notations

$\Delta x_t = x_t - x_{t-1}$	The difference of the value of x between two consecutive timesteps
$x_t^{(i)}$	The value x associated with the agent i at the timestep t
$\varepsilon = \frac{1}{2}\epsilon_{sym} + \epsilon_{asym}$	The normalized tweak applied by the LP to the mid-price, relative to current ECN prices.

2.2 Definitions

Prices: The LP quotes bid and ask *prices* at which it is willing to buy and sell, respectively. The ECN also exposes *prices* on both the bid and ask sides. The best bid (respectively ask) corresponds to the highest (lowest) price at which a bid (ask) order can be executed on the ECN. We call *mid-price* P_{mid} the ECN mid-price; i.e the average of the best bid and best ask prices in the limit order book of the ECN. The mid-price is often use as a price of reference and the LP typically quotes their prices relative to P_{mid}.

Market Spread: The *market spread* is the difference between the best ask price and the best bid price on the ECN LOB. We often use the half market spread s^{ref} to reference the spread on each side. We can therefore write:

$$P_{ask}^{ref} = P_{mid} + s_{ask}^{ref}; \qquad P_{bid}^{ref} = P_{mid} - s_{bid}^{ref};$$

Spread: In the context of the LP, the *spread* corresponds to the difference between the price quoted by the LP and the mid-price. In a similar fashion than above we can write the prices quoted by a LP i as a function of the spread:

$$P_{ask}^{(i)} = P_{mid} + s_{ask}^{(i)}; \qquad P_{bid}^{(i)} = P_{mid} - s_{bid}^{(i)}; \tag{1}$$

We note however, that the spread $s_{ask/bid}^{(i)}$ is at the discretion of the LP, and different market making strategies will yield different spread. Depending of the quantity the LT wants to trade, the spread might differ.

Inventory: To accommodate the limited supply and demand, the LP typically maintains an inventory of the quantity traded until an investor accept to trade on the side at which it holds inventory. A

[1] First In First Out

positive inventory indicates that the LP has bought more than it has sold and vice versa. As presented in [1] and [12], the *inventory* plays an important role in the pricing of the LP.

Risk aversion: The risk aversion of an agent corresponds to a regularizer applied to the PnL, so as to penalize its fluctuations associated with carrying a large inventory. We choose to model the risk aversion as a penalty on the Inventory PnL, to capture the price risk the agents bear when holding inventory ([12]).

Skewing: With the aim of disposing of its inventory, the LP can make its prices on the side at which it holds inventory, more appealing to the LTs. The action of publishing asymmetric prices with one side more attractive than the other is called *skewing*. The intensity of the skew is linked with the level of *risk aversion* of the LP.

Hedging: Another way for the LP to reduce its inventory is to *hedge*, i.e perform a trade which reduces its inventory. The LP takes the role of the LT in this configuration and therefore have to pay a cost.

Spread PnL: This excess in price between the bid and ask prices and the mid-price is paid by the LT. It can be seen as the cost the LT pays in order to trade the security. The profit or loss made by the LP by facilitating a trade is called the *spread PnL*. It is a function of the quantity traded q and the *spread* at which the order was executed $s^{(i)}(q)$:

$$\mathrm{PnL}_{\mathrm{spread}} = q \cdot s^{(i)}(q)$$

Inventory PnL: As we have explained, holding inventory comes with profit or loss because of the frequent fluctuation of the mid-price. The *inventory PnL* will be affected by the quantity of inventory q and by the mid-price move ΔP_{mid}:

$$\mathrm{PnL}_{\mathrm{inventory}} = q \cdot \Delta P_{mid}$$

Total PnL: The total PnL also simply referred as PnL, is the sum of *Inventory PnL* and *Spread PnL*.

$$\mathrm{PnL} = \mathrm{PnL}_{\mathrm{inventory}} + \mathrm{PnL}_{\mathrm{spread}}$$

3 MULTI-AGENT FRAMEWORK

3.1 Configuration and Training

To conduct our research we consider the dealer market as a *Partially-Observable Stochastic Game* defined in [15]; where each agent has only a partial view of the state of the world and aims at maximizing its own reward function by interacting with its environment. The different families of agents in our system are LP, LT and ECN. At each timestep, the LP updates the prices that it streams to the LT according to its observation of the market. Subsequently, and based on this new observation, the LT decides whether or not to trade with the LP or the ECN. The LP can also choose to hedge its position by trading with the ECN. The problem has thus been designed as a Stackelberg game, where the LT *reacts* to the action that the LP takes (setting the prices it is willing to trade at).

As the focus of this paper lies in the interaction between the RL based LPs and LTs, we purposely give a brief description of the ECN model used in our experiments and leave a more extensive modeling for further research. Inspired by the work of Cont and Muller [7], we developed a statistical model to simulate the evolution of the order book as a function of its current state. The ECN agent is therefore able to react to the orders executed by the LPs and LTs and adjust its internal state to replicate the dynamics of the market. Our model is composed of three multivariate Gaussian mixtures calibrated using L2-level order book data (i.e snapshots of the ECN order book), and predicting the variation of the order book's snapshot over an interval of time dt. The first model is used to sample the initial snapshot of the book, that is, how much volumes is available at each level of the book. The second mixture models the variation of volume for each level as a function of the current volume. Finally, the third model decomposes the variation of volume to apply, into multiple smaller orders that will be added to the ECN.

3.2 Shared Policy

The problem of multiple learning agents evolving in a decentralized partially-observable Markov decision process (DEC-POMDP) is known to be challenging [3]. Despite sharing the same environment, the agents have a different set of observations and act individually to maximize their own reward function. The already complex nature of the problem and the introduction of yet another set of learning agents to represent the Liquidity Takers can raise the question of the scalability of our approach and the training efficiency. We leverage the Centralized Training with Decentralized Execution technique presented in [16] to train two shared policies for the two families of agent present in our system. The use of a shared policy not only allow us to dramatically reduce the complexity of the problem by only having to train two neural networks; but also helps generalizing the trained policy by allowing the interpolation over the full behavior space of the agent family.

Within a family of agent Λ, each agent i is of a specific type $\lambda^{(i)}$ characterizing the vector of parameters used by the reward function of the agent and defining its behavior. They all share the same action space $\mathcal{A}$ but in order to maximize their own cumulative reward R_i, they may take different actions $a_t^{(i)}$ due to their different states $s_t^{(i)}$ at any given time t and their different reward function. The common policy π_θ^Λ shared by all the agents of the family Λ is conditioned on the agent type $\lambda^{(i)}$ in order to allow generalization over the entire behavior space.

The policy is trained for a large number of episodes until convergence. The type $\lambda^{(i)}$ is sampled from a predefined distribution at the beginning of each episode and for each agent i. It is added to the observations of the agent as a constant value throughout the episode. The probabilistic nature of the agent type attribution enable the policy to learn across the spectrum of agent types.

3.3 Liquidity Provider Agent

The role of the Liquidity Provider is to facilitate trading by providing liquidity to the other market participants. They continuously publish prices at which they are ready to buy and sell.

3.3.1 *Reward Function.*

We model the Liquidity Provider family as a spectrum of behaviors where on one end the agent tries to maximize its profit and on the other end it tries to gain a targeted share of the market. We propose a parametrized reward function able to define under a single formulation, a range of behaviors forming the Liquidity Provider family.

More rigorously the per timestep reward of a Liquidity Provider agent i can be formulated as follows:

$$\text{Reward}_t = w \cdot \alpha \cdot \text{PnL}_t(\gamma) - (1-w) \cdot \text{MarketShare}_t(m^*)$$

The parameter w acts as a weight defining the importance allocated to the PnL or the the market share objective. An agent with $w = 1$ will only trade with the aim to maximize its PnL. On the other hand, a weight $w = 0$ will define an agent whose prime objective is to meet the market share objective irrespective of its PnL. The parameter α acts as a normalizer on the PnL component to make it comparable with the MarketShare objective.

The **PnL** component at an instant t is defined by (2). The second part of the equation represents the risk aversion penalty associated with the inventory held. We call γ the risk aversion of the LP.

$$\text{PnL}_t^{(i)}(\gamma) = \Delta\text{PnL}_t^{(i)} - \gamma\left|\Delta\text{PnL}_{\text{inventory},t}^{(i)}\right| \quad (2)$$

The **Market Share** component of the reward function is defined in (3). The goal is to minimize the difference between the empirical market share of the LP at the instant t and the target m^* specified as a parameter.

$$\text{MarketShare}_t^{(i)}(m^*) = \Delta\left|\frac{\sum_{s=0}^{t-1} m_s^{(i)}}{\sum_{j=0}^{N}\sum_{s=0}^{t-1} m_s^{(j)}} - m^*\right| \quad (3)$$

This parametrized formulation provides enough flexibility to represent different behaviors. We can represent agents who want to increase their footprint in the market thanks to the *Market Share Objective* component or the LPs who mainly trade to generate PnL with different level of risk aversion, or even a combination of both. By using the same reward formulation for all the LP agents and by incorporating the parameters of the reward function in the observation space, we train a *shared policy* capable of interpolating over the behavior space of the Liquidity Provider.

3.3.2 Pricing Formulation.

Market making is often seen as an optimal control problem ([2], [12]) where the LP aims at finding the optimal prices to send to the LT in order to maximize its reward function. The LP quotes relative to the mid price and the problem then becomes finding the optimal spread to apply on the bid and ask sides. To model the spread applied by the LP agent, we decompose (1):

$$P_{ask} = P_{mid} + (s_{ask}^{ref} + s_{bid}^{ref})\left[\frac{1}{2}(1+\epsilon_{sym}) + \epsilon_{asym}\right] \quad (4)$$

$$P_{bid} = P_{mid} - (s_{ask}^{ref} + s_{bid}^{ref})\left[\frac{1}{2}(1+\epsilon_{sym}) - \epsilon_{asym}\right] \quad (5)$$

The two newly introduced parameters ϵ_{sym} and ϵ_{asym} will be learnt by the policy network as a function of the observations available to the RL agent. As we can see in (4) and (5), ϵ_{sym} is a *symmetric* tweak of the prices around the mid-price, it can be thought of as the parameter controlling the willingness of the LP to win the flow. A negative ϵ_{sym} indicates that the LP offers to trade at a price more favorable for the LT, generating less Spread PnL for the LP but with a higher probability of being executed. ϵ_{asym} is an *asymmetric* tweak and is used for skewing (i.e making one side more attractive than the other to investors) with the objective of reducing the inventory held.

3.3.3 Agent Configuration.

Actions: At each timestep, the RL agent characterizing a LP can decide the value for three parameters:

- Symmetric price tweak ϵ_{sym}
- Asymmetric price tweak ϵ_{asym}
- Fraction h of its inventory to hedge

Observations: For the agent to learn the optimal policy the following observations are passed to the policy network at each timestep:

- Reference mid-price P_{mid} and its history
- Inventory z currently held
- Fraction of time elapsed since the start of the simulation
- Empirical Market Share m the LP had in the previous timestep
- The top n levels of the ECN order book
- Cost associated with hedging for multiple hedge fractions h

The parameters characterizing the agent type are also added to the observations:

- Weight w associated with the PnL component
- Risk aversion parameter γ
- Market Share target m^*
- Empirical probabilities of being connected with a LT and a ECN agent

3.4 Liquidity Taker Agent

The Liquidity Taker, also called investor is an actor who enters the market to trade. As opposed to the Liquidity Provider, it doesn't publish quotes; rather it consumes the prices streamed by the LP and makes the choice on whether or not they wish to trade with it, but do not have any obligation to do so. We innovate by introducing a RL-based agent to take the role of the Liquidity Taker in our market simulator. We start by formalizing the Liquidity Taker reward function and present a toy example demonstrating the different behaviors a LT can adopt.

3.4.1 Reward Function.

Similarly to the LP, the behavior a LT adopts can be represented as a combination of two components: a PnL component and a Target Flow component. The PnL part is analogous to the one presented for the LP, the agent aims at maximizing its PnL and execute trade optimally for that purpose (Informed traders are a typical example of PnL driven LT where the PnL component predominates). On the other end of the spectrum, some LT are more flow driven. Their primary objective is to trade a targeted quantity. The target Flow indicates how much they should trade on each side. To reduce the complexity of the LT agent, we assume that it always trades a fixed unit quantity. This simplifies the Flow objective component to only track the frequency for each action. The formulation also supports mixed behavior where the agent tries to satisfy its trade objective with a certain tolerance, allowing it to generate more PnL.

The formulation below defines a family of agents where a given agent i could fall anywhere on that spectrum. The parameter w is used as weight to make the agent more "PnL" or "Flow" driven.

$$\text{Reward}_t = w \cdot \alpha \cdot \text{PnL}_t(\gamma) - (1-w) \cdot \text{Flow}_t(\mathbf{q}^*)$$

The **PnL** component is the same as the one presented in (2) where a penalty is added to the PnL in order to accommodate for the risk associated with holding inventory:

$$\mathrm{PnL}_t^{(i)}(\gamma) = \Delta\mathrm{PnL}_t^{(i)} - \gamma\left|\Delta\mathrm{PnL}_{\text{inventory},t}^{(i)}\right| \quad (2)$$

The **Flow** objective of the reward function evaluates the distance between the flow executed by the agent up until t and the flow objective $\mathbf{q}^*$ specified as parameter. In practice, the objective is an array containing the targets for each of the the possible actions a of the action space $\mathcal{A}$.

$$\mathrm{Flow}_t^{(i)}(\mathbf{q}^*) = \frac{1}{|\mathcal{A}|}\sum_{j\in\mathcal{A}}\Delta\left|\frac{1}{t}\sum_{s=0}^{t-1}\mathbb{1}_{\{a_t^{(i)}=j\}} - q_j^*\right| \quad (6)$$
$$\sum_{j\in\mathcal{A}} q_j^* = 1$$

To illustrate the variety of behaviors of this type of agent, we present in *Figure 1* a toy example where an artificial trend is added to the mid-price. Adding this trend allows us to easily identify the best trading strategy the agent should learn in order to maximize its profit. The idea is to observe how the behavior of the LT agent evolves as the weight w applied to the PnL component changes. For this example we use a simple configuration of our simulator with one LP and one LT. We parametrize the LT agent with no risk aversion $\gamma = 0.0$; and with an objective to sell 25% of the time and buy 75% of the time: $q_{ask}^* = 0.25$; $q_{bid}^* = 0.75$.

As we can observe in *Figure 1a* when the principal objective for the LT is to meet the specified flow objective, the agent trades with no particular consideration of the price. It successfully buys 75% of the time and sells 25%. With the weight on the PnL component increasing the agent learns to adapt its trading strategy to buy and sell when it is the most favorable to generate PnL. For $w = 1$ in *Figure 1d*, the LT learnt to trade perfectly following the famous motto "buy low, sell high". Intermediate behaviors are shown in *Figure 1b, 1c* where we see that the agent gradually discard its objective of matching its flow targets and focus more on maximizing its PnL.

3.4.2 Agent Configuration.

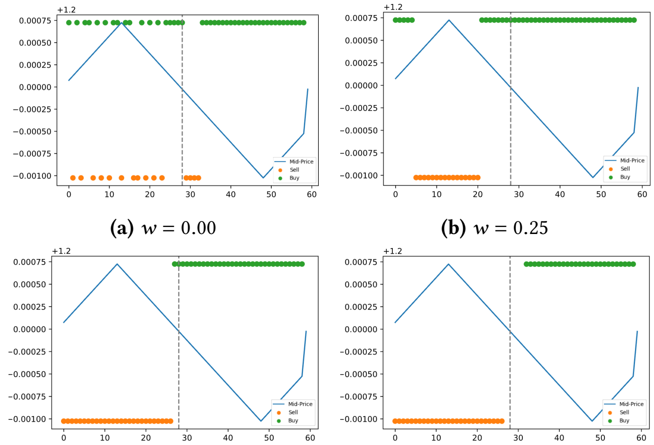
Actions: At every timestep, the RL-based Liquidity Taker agent can choose to *buy*, *sell* or *not trade* in order to maximize its reward function.

Observations: At each timestep, the policy network of the LT agent is fed with the observations below:

- Reference mid-price P_{mid} and its history
- Inventory z currently held
- Fraction of time elapsed since the start of the simulation
- Proportion of the flow the LT has traded on each side q_{ask} and q_{bid}
- Cost associated with executing a trade on both side for each LP the LT is connected to

We also enrich the observations with the parameters of the reward function and the connectivity of the agent type for the shared policy to learn the distribution of behaviors characterizing the LT:

- Weight w associated with the PnL component
- Risk aversion parameter γ
- Flow targets q_{ask}^* and q_{bid}^*



(a) $w = 0.00$ **(b)** $w = 0.25$

(c) $w = 0.75$ **(d)** $w = 1.0$

Figure 1: Spectrum of behavior for the Liquidity Taker
With this simple example where we have artificially added a trend in the mid-price, we see the different behaviors the LT can adopt depending on its reward function. As the weight w associated with the PnL component increases, the LT progressively trades more and more optimally to maximise its PnL at the expense of its target flow of 75% Buy-25% Sell.

- Empirical probabilities of being connected with a LP and a ECN agent

4 EXPERIMENTS

We focus our attention on the empirical study of the effect that the different behaviors of LT can have on the LP's actions. We aim at answering the following questions with our study: 1) Can the two groups of agents learn simultaneously? 2) How does the LP agent adapt its behavior in the presence of different types of LT? 3) Do we need this new type of agents to replicate observed market properties?

We present below the results from various experiments trying to approach the problem from different angles. In the first configuration of experiments, we change the proportion of LT actors of a certain type in the system. We start with a similar configuration than the one presented in [10] and gradually increase the number of PnL driven investors. We try to evaluate how the heterogeneity of behaviors can affect the LP's actions. The second configuration of experiments takes the approach of varying the connectivity between the agents of a particular type. The study of the connectivity is important because two agents not connected have still an indirect effect on each other by interacting with and modifying the environment.

Using the RLlib library [13] and the OpenAI Gym framework [4] we developed a multi-agent market environment, allowing us to put in competition different types of RL based market participants. The shared policies of the agents are trained using the standard RLLib implementation of the Proximal Policy Optimization (PPO) algorithm which supports parallel execution providing us with the scalability needed for a complex environment like ours.

4.1 The Impact of Diversity

4.1.1 Price Tweak Distribution.

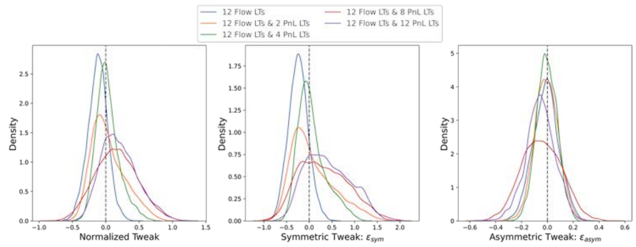


Figure 2: Distribution of the price tweak

Setup: We present in *Figure 2* the distribution of the normalized tweak ε applied by the LP to its prices in order to attract LT. As presented in 3.3.2 (4) and (5), the spread applied to the ECN mid price have a symmetric ϵ_{sym} and an asymmetric tweak ϵ_{asym}, we therefore also plot the distribution for these two components. We ran multiple experiments varying the proportion of Flow and PnL driven LTs in the system. The experiments are run with 3 LPs, 12 Flow driven LTs and an increasing number of PnL driven LTs (0, 2, 4, 8, 12). The goal of these experiments was to understand the effect of the proportion of different LT types on the way the LPs price. A *negative* normalized tweak ε, indicates that the LP is offering better prices than the ECN and the more negative the tweak is, the more attractive the prices will be to investors. For the asymmetric component of the tweak ϵ_{asym}, a *negative* value indicates that the *ask* side is favoured and a *positive* value favours the *bid* side.
Analysis: As the proportion of PnL driven LTs increases, we see a shift in the distribution of the normalized tweak ε towards higher values indicating that the LP published worse prices. Looking at the two components of the tweak, we observe that this shift comes from the symmetric component ϵ_{sym}. The LP's PnL suffers when too many informed LTs are in the system. To compensate its losses, **the LP becomes more conservative in its pricing as the proportion of PnL driven LTs increases**. The asymmetric tweak ϵ_{asym} is centered around the origin implying that, as expected, the proportion of PnL driven LTs has no impact on the direction of the skew. However, we do observe fatter tails for this distribution suggesting an increased skewing intensity from the LPs on one side or the other, which can also be interpreted as a **higher risk aversion**. Since both the LPs and the PnL driven LTs try to maximize their Inventory PnL, as the proportion of PnL LTs increases the adversarial effect on the LPs intensifies and their Inventory PnL get lower.

4.1.2 Flow by LT Agent type.

Setup: In *Figure 3* we plot the average flow between a LP and the LT of different types (Note that both the bid and ask sides are combined on the same chart). The reward function of the PnL driven LT being different than the reward function of the Flow driven LT, they should adopt different behaviors and have different reactions to the way the LP tweaks its prices. We expect the PnL driven LT to be more demanding and only trade for good prices. On the other hand, because their goal is to meet a flow objective, the Flow driven LT

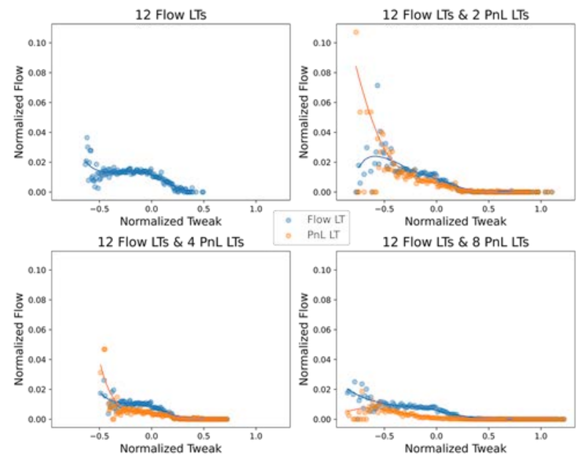


Figure 3: Flow by LT type

needs to trade with little respect to the price. For better prices we should therefore see a higher flow coming from the PnL driven LTs. We run multiple experiments with different proportions of PnL and Flow LTs.
Analysis: As we can see in the top left quadrant of *Figure 3*, a configuration with only Flow driven LT does not help replicating the "stylized facts" observed in the market. The flow between the LP and the LT increases slowly as the prices get better for $\varepsilon \in [0.25, 0.00]$. With $\varepsilon \geq 0.25$, the prices streamed by the LP are so bad that it doesn't attract any flow and the LT prefer to trade with the ECN instead as it offers better prices. We observe that with $\varepsilon \leq 0.00$, the flow plateaus with a slight uptake for very good prices. With the introduction of PnL driven LTs, we can see in the top right quadrant, what we were expecting: because they only trade when the price is best for their PnL, **the PnL driven LTs see an exponential uptake of their flow as the prices improve**. However, as we can observe in the bottom charts, as we increase the number of PnL driven LTs the convexity in the flow uptake shrinks and with too many informed LTs, their flow becomes almost null. This is associated with the results presented in 4.1.1 where too many PnL LTs make the LP tweak its prices less, not making it worthwhile for the PnL LT to trade.

4.1.3 PnL by Agent type.

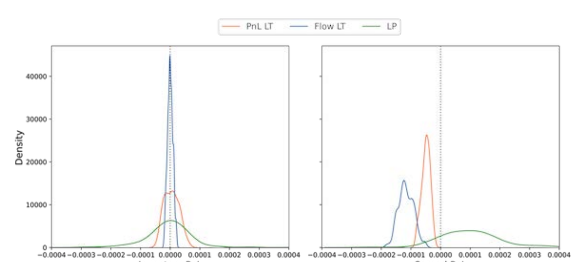


Figure 4: Spread and Inventory PnL by Agent type

Setup: For the configuration of 12 Flow and 2 PnL driven LTs, we now try to gain more insights about the different types of agents. We present in *Figure 4* the average Spread and Inventory PnL for each type. Looking at the PnL helps us understand their trading pattern. We wish to understand how each agent type performs against the others. As we have explained previously the Flow and PnL LTs use a shared policy learning a family of reward functions. The LPs use a shared policy of their own, which means that we are in a situation where two shared policy are competing. We recall that when the LT trades with a LP, it pays the *spread*, generating a positive Spread PnL for the LP but negative for the LT. In these experiments the flow objective of the Flow driven LT is to trade 50% of the time on the bid side and 50% on the ask side.

Analysis: Looking at the Inventory PnL in *Figure 4*, we see that the distribution of PnL for all 3 agent types is centered around the origin. The distribution for the Flow LT is narrower than the others because of their objective to trade frequently. Consequently they do not hold a large inventory and therefore generate less Inventory PnL. As expected the PnL driven LT has a wider distribution of Inventory PnL indicating that it earns more PnL by strategically holding inventory. However, the fact that it is only connected to the ECN and the LPs, doesn't allow the PnL LT to generate as much Inventory PnL than the LP. The LP benefits from its connectivity with the Flow LTs providing the flow required to build inventory. The analysis of the Spread PnL distribution also offers interesting insights. As we were expecting, because the LT pays the spread, its Spread PnL is negative. The Spread PnL of the LP is mostly positive but is sometimes negative as the LP pays the spread when it hedges itself at the ECN. The Spread PnL of the PnL driven LT is significantly higher than the Spread PnL of the Flow LT. The PnL LT agent learns to trade with the objective of maximizing its PnL, it therefore tries to trade only when the Spread is more favourable, generating a higher spread PnL.

4.2 Role of connectivity with Flow LTs

In the next set of results presented below, we take a different approach and play with the connectivity between the Flow LTs and the LP. Since an agent can only trade with a connected agent, its connectivity with the other families implicitly have an effect on the agent's behavior and thus is part of its type. Because of their nature, the Flow driven LT, have to trade to meet their flow objective irrespective of the price. That makes them a guaranteed source of flow for the LP and therefore a reliable source of Spread PnL. By varying the connectivity between these two types of agent and in the presence of PnL driven LTs, we wish to understand how the behavior of the LP is affected.

4.2.1 Pricing as a function of connectivity.

Setup: In *Figure 5*, we display the distribution of the tweak applied by the LP in its prices as a function of the empirical probability of being connected to a Flow LT. As before, the lower the price tweak, the more attractive the prices will be for LTs. The goal of this experiment is to understand how the LP adapts its behavior when its guaranteed source of flow is limited because of a lower probability of being connected.

Analysis: A clear trend appears on *Figure 5*, as the probability of being connected decreases the LP becomes more conservative in its prices. In fact, with low connectivity with Flow driven LTs, it gets more and more difficult to dispose of its inventory. Consequently, to accommodate for the risk associated with holding inventory the LP publishes worse prices. We observe a similar effect than in 4.1.1, indicating that reducing the connectivity between LP and Flow LTs yield the same behavior than having a higher proportion of PnL driven LTs.

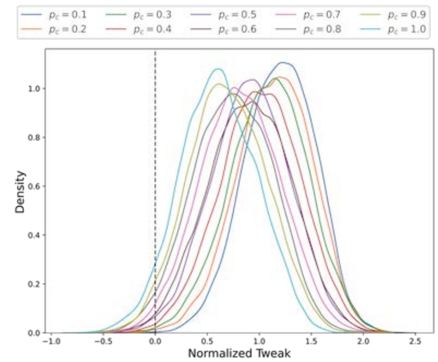


Figure 5: Distribution of the price tweak for different level of connectivity between Flow LTs and LPs

4.2.2 Skew intensity.

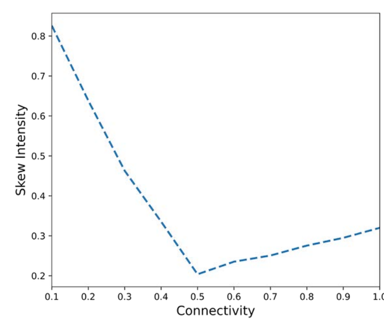


Figure 6: Skew intensity as a function of Connectivity

Setup: We now study the effect of connectivity on the intensity of the skew the LP applies to its prices, that is how much the LP tweak its prices asymmetrically to attract the investors on the side at which it holds inventory in order to reduce its risk. We show on *Figure 6*, the skew intensity as a function of the connectivity. With a lower probability of being connected with a Flow LT, it will be more challenging for the LP to manage its inventory because of the absence of flow guarantee the Flow LT provides. The LP is thus exposed to an unfavorable price move that would impact its Inventory PnL. We therefore expect the LP to be more willing to get rid of its inventory for lower connectivity which would be materialized by a higher skew in its prices.

Analysis: As we can see in *Figure 6*, the probability of being connected to a Flow LT affects the way the LP prices. **With low connectivity, the LP is more aggressive in its skew**. The incertitude about the ability to dispose of its inventory is forcing the LP to skew more its prices to increase the chances of attracting investors.

As the connectivity increases, the LP reduces the intensity of the skew linearly up until a probability of 0.5 of being connected, after which the intensity remains relatively stable until full connectivity.

4.3 Risk aversion

In this last experiment, we evaluate the behavior of the LP with different level of risk aversion. We keep the configuration of LTs constant and only change the risk aversion factor γ of the reward function of the LP.

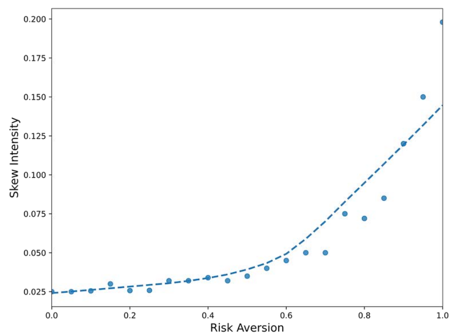


Figure 7: Skew intensity as a function of Risk Aversion

Setup: We present in *Figure 7*, the skew intensity for different level of risk aversion. The goal of this experiment is to understand if the LP learns to skew more or less its prices based on its type. A more risk averse LP would feel the urgency to exit its position to be less exposed to an unfortunate price move. It will therefore skew mode in order to have higher chances to attract investors and get rid of its inventory.

Analysis: As we can see in *Figure 7*, for a higher level of risk aversion the LP learns to skew more. We observe two regimes, the first one ranging from 0.0 to 0.6 where the skew intensity grows linearly as a function of the risk aversion. After, the critical point of 0.6, we observe a more intense effect of the risk aversion in the skewing of the LP. **The shared policy of the LPs has then successfully learnt to adapt its pricing strategy as a function of its risk aversion parameter.** By strategically updating its prices, the LP can manage its inventory efficiently in order to maximize its PnL without being exposed to the price risk.

5 CONCLUSIONS AND FUTURE WORK

With our new unified RL formulation for both the LP and the LT, via a parametrized family of reward functions expressing the trade-off between PnL and a targeted quantity (specific to each family). The RL-framework we propose and the use of shared policies enable the generalization and the interpolation over a wide range of behaviors for each group of agent. We were able to study the behavior of the different participants in a market simulator composed of a variety of financial agent types providing a more representative setting of the real life market.

The simultaneous learning of the shared policies for two competing groups of RL-based agents raises some interesting questions from a game-theoretic perspective and could be the object of more in-depth future research.

DISCLAIMER

REFERENCES

[1] Yakov Amihud and Haim Mendelson. 1980. Dealership market: Market-making with inventory. *Journal of Financial Economics* 8, 1 (1980), 31–53. https://doi.org/10.1016/0304-405X(80)90020-3
[2] Marco Avellaneda and Sasha Stoikov. 2008. High-frequency trading in a limit order book. *Quantitative Finance* 8, 3 (2008), 217–224. https://doi.org/10.1080/14697680701381228 arXiv:https://doi.org/10.1080/14697680701381228
[3] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. 2000. The Complexity of Decentralized Control of Markov Decision Processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (Stanford, California) *(UAI'00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 32–37.
[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
[5] Nicholas Tung Chan and Christian Shelton. 2001. An electronic market-maker. (2001).
[6] Tung Chan. 2001. *Artificial markets and intelligent agents.* Ph.D. Dissertation. Massachusetts Institute of Technology.
[7] R Cont and MS Mueller. 2021. A stochastic PDE model for limit order book dynamics. *SIAM Journal on Financial Mathematics* (2021).
[8] Wei Cui and Anthony Brabazon. 2012. An agent-based modeling approach to study price impact. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*. 1–8. https://doi.org/10.1109/CIFEr.2012.6327798
[9] Sanmay Das. 2003. An agent-based model of dealership markets. In *Proceedings of the International Workshop on Complex Agent-based Dynamic Networks, Oxford.*
[10] Sumitra Ganesh, Nelson Vadori, Mengda Xu, Hua Zheng, Prashant Reddy, and Manuela Veloso. 2019. Reinforcement learning for market making in a multi-agent dealer market. *arXiv preprint arXiv:1911.05892* (2019).
[11] Mark B. Garman. 1976. Market microstructure. *Journal of Financial Economics* 3, 3 (1976), 257–275. https://doi.org/10.1016/0304-405X(76)90006-4
[12] Olivier Guéant, Charles-Albert Lehalle, and Joaquin Fernandez-Tapia. 2013. Dealing with the inventory risk: a solution to the market making problem. *Mathematics and financial economics* 7, 4 (2013), 477–507.
[13] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. 2018. RLlib: Abstractions for Distributed Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 3053–3062. http://proceedings.mlr.press/v80/liang18b.html
[14] Thomas Spooner and Rahul Savani. 2020. Robust Market Making via Adversarial Reinforcement Learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4590–4596. https://doi.org/10.24963/ijcai.2020/633 Special Track on AI in FinTech.
[15] Justin K. Terry, Nathaniel Grammel, Benjamin Black, Ananth Hari, Caroline Horsch, and Luis Santos. 2020. Agent Environment Cycle Games. *CoRR* abs/2009.13051 (2020). arXiv:2009.13051 https://arxiv.org/abs/2009.13051
[16] Nelson Vadori, Sumitra Ganesh, Prashant Reddy, and Manuela Veloso. 2020. Calibration of Shared Equilibria in General Sum Partially Observable Markov Games. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 14118–14128. https://proceedings.neurips.cc/paper/2020/file/a2f04745390fd6897d09772b2cd1f581-Paper.pdf

[17] Svitlana Vyetrenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Dervovic, Manuela Veloso, and Tucker Hybinette Balch. 2019. Get Real: Realism Metrics for Robust Limit Order Book Market Simulations. arXiv:1912.04941 [q-fin.TR]

[18] Elaine Wah and Michael P. Wellman. 2015. Welfare Effects of Market Making in Continuous Double Auctions. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems* (Istanbul, Turkey) *(AAMAS '15)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 57–66.