

Multi-Agent Trading Simulation with Classical and Hybrid Agents under Adversarial Market Dynamics

Anuj Gupta¹, Apurva Mishra², Mradul Bhardwaj³, Dr. Santosh Sahu⁴

Associate Professor¹, Research Scholar², Assistant Professor³, Assistant Professor-3⁴
Department of ECE¹, Department of Law², Department of Commerce³, Department of
Computer Science and Engineering (CSE)⁴
Chandigarh University, NH-05, Ludhiana - Chandigarh NH, Chandigarh State, Punjab
140413^{1,2,3}

School of Computer Science and Engineering, Galgotias University, Greater Noida UP⁴

Corresponding email: annul1gupta@gmail.com

Abstract—Financial markets are heterogeneous complex adaptive systems made up of heterogeneous interacting agents whose systemic behavior dictates price formation and systemic stability. As the influence of algorithmic and AI-based trading systems grows, it is now of paramount importance to learn the dynamics of emergence in the interaction between agents in competition. The present paper suggests the multi-agent trading simulation model that combines classical agents based on rules, hybrid agents based on deep learning and reinforcements learning, as well as adversarial agents in the same market setting. The models are endogenous price formation, liquidity dynamics and adversarial manipulation, and are used to analyze emergent volatility, market and systemic risk. As experimental results show, hybrid agents are more profitable than classical strategies, but increase volatility when adversarial pressure rises. Stability can be greatly increased by including risk- aware reward shaping. The suggested framework offers a systematic approach to the assessment of AI-based financial ecosystems and how they resist during adversarial situations.

Keywords— Multi-agent systems, algorithmic trading, reinforcement learning, adversarial dynamics, market stability, deep learning.

1. INTRODUCTION

The algorithmic and high-frequency trading systems have become very common in the structure of the modern financial markets [1], [2]. These autonomous agents are continuously interacting to generate nonlinear feedback processes which determine price discovery and the provision of liquidity. Although algorithmic trading has increased efficiency it has been identified to cause episodic instability like the Flash Crash [3].

The conventional models of financial theories in equilibrium cannot explain emergent phenomena, including volatility clustering and fat tails of returns [4]. An alternative paradigm is Agent-Based Computational Finance (ABCF), which is a model of markets as complex adaptive systems of heterogeneous agents [5], [6].

Recent progress in the field of deep learning and reinforcement learning has facilitated dynamic decision-making adaptive trading systems [7], [8]. The hybrid models of predictive neural networks with reinforcement learning have shown enhanced trading performance [9], [10]. Nonetheless the systemic risks that come with large-scale implementation of learning agents are synchronization of behaviour and amplification of feedback [11].

Moreover, antagonistic trading practices like liquidity manipulation and spoofing can also shape market forces [12], [13]. Financial system resilience depends on learning to interact with classical and AI-oriented agents in times of adversarial stress.

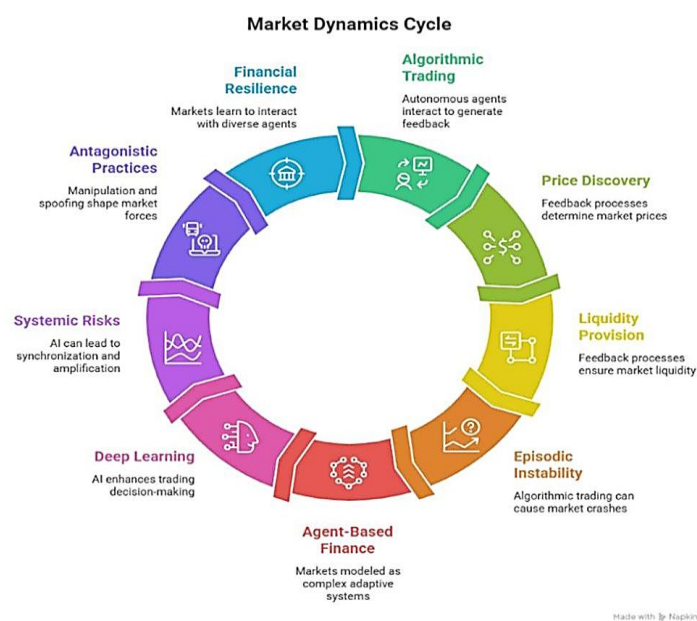


Fig. 1. Market Dynamics Cycle

Contributions:

The contributions made in this paper are as follows:

1. Offers a multi-agent trading simulation that is propose a multi-agent trade that is a unified classical, hybrid and adversarial simulation.
2. Models are endogenous price formation by imbalance of demand and supply.
3. Proposes a Market Stability Index (MSI) to evaluate the systemic analysis.
4. Studies emergent behavior and stability at adversarial dynamics.
5. Demonstrates the stabilizing effect of risk-aware reinforcement learning

2. RELATED WORK

Empirical market stylized facts have been found to be replicated using agent-based financial models [5], [14] where heterogeneous interactions are used. The Santa Fe Artificial Stock Market has shown that boundedly rational agents create realistic price movements [15].

Trading and portfolio management have to be reinforced by learning [7], [16]. Deep Q-Networks (DQN) allow making decisions sequentially in case of uncertainty [8]. Financial time series LSTM networks are efficient deep learning models that can capture financial time series temporal dependencies [17], [18].

Such antagonistic interactions in the financial markets have been examined in terms of toxic order flow and market manipulation [12], [19]. Nevertheless, not many studies involve adversarial agents as a component of a competitive multi-agent AI trading scenario.

3. METHODOLOGY

The suggested framework describes financial markets as a competitive multi-agent ecosystem where the heterogeneous trading agents play in a simulated environment. The architecture is based on agent-based computational finance, which models that complicated market behaviour arises not out of centrally-determined equilibrium, but rather out of local-level agent interactions [5], [6], [14]. The methodology consists of five core components:

1. Market Environment Design
2. Agent Architecture Modeling
3. Risk-Aware Adaptation and Learning
4. Adversarial Stress Modeling.
5. Stability and Emergent Behavior Evaluation.

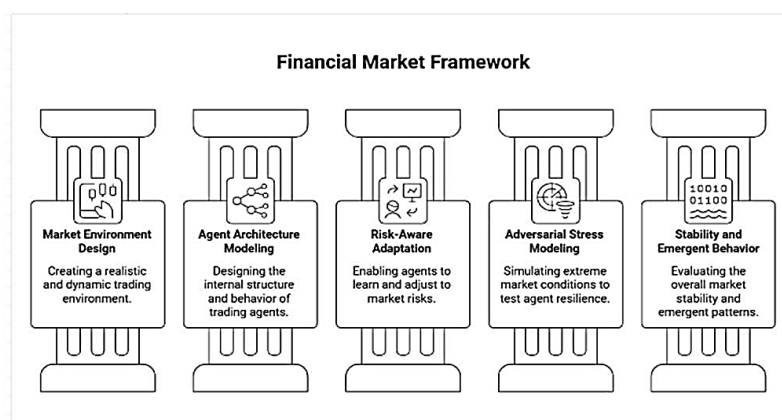


Fig. 2. Financial Market Framework

A. Market Environment Design

The market is modeled as a discrete time simulation environment in which various agents monitor the market environment and make trading decisions during each timestep.

The environment includes:

- Asset price
- Historical price window
- Trading volume
- Realized volatility
- Order imbalance

- Liquidity depth

The formation of prices is a result of the aggregate demand and supply among the interested agents, which is in line with the modelling strategies of microstructures [1], [12]. Buy pressure and sell pressure are correlated such that when the former is higher than the latter, prices are high and the opposite is true. Exogenous uncertainty, in addition to macro-level information shocks is simulated by adding random stochastic noise [4].

This is in contrast to the equilibrium-based pricing models, which only measures the endogenous feedback between the trading agents, which is one of the central features of the agent-based financial modeling [5], [6].

The simulation has parameters that can be configured such as:

- Agent composition ratios
- Sensitivity of market on order imbalance.
- Liquidity constraints
- The levels of adversarial intensity.

This is flexible allowing experimental comparison between various market arrangements in a controlled fashion.

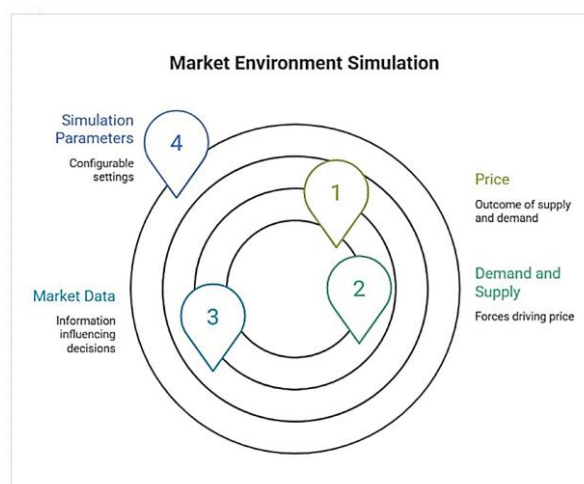


Fig. 3. Market Environment Simulation

B. Agent Architecture Modeling.

The agents are divided into three classes:

1. Classical Rule-Based Agents
2. Hybrid AI-Based Agents
3. Adversarial Agents

Such a heterogeneous arrangement corresponds to the heterogeneous agency models of finance [11], [15].

1. Classical Rule-Based Agents

The classical agents employ deterministic technical strategies that are commonly applied in conventional algorithmic trading systems [1].

Strategies include:

- Moving Average Crossover
- Relative Strength Index (RSI)
- Momentum-Based Trading

These agents work off of historical price indicators and monotonic levels. They are not flexible and responsive to market feedback. These rule-based agents provide a consistent point of reference when analysing the learning-based systems [2].

Since classical agents are not dynamic in updating strategies, they provide a baseline liquidity although they could respond in a predictable way to market fluctuations.

2. Hybrid AI-Based Agents

Hybrid agents are predictive modeling and reinforcement learning to allow an adaptive trading behavior.

a. Price Prediction Module

A LSTM neural network is a neural network that is the model of temporal relationship in the financial time-series data. The LSTM network is commonly known to be highly effective in modelling long-term dependencies in sequential data [17], and has been effectively used in financial market prediction [18].

This element gives future projections in addition to mere historical averaging.

- Historical price sequences
- Volatility patterns
- Technical indicators

This component provides forward-looking signals beyond simple historical averaging.

b. Decision Module in Reinforcement Learning.

The second element is a Deep Q-Network (DQN), which allows the agents to choose trading policies depending on the perceived state data. Reinforcement learning with DQN has shown great productivity in sequence decision-making tasks [8].

The reinforcement learning concept has been utilized in the domain of financial trading to address the issue of trade execution and portfolio management [7], [16].

The module of decision takes into account:

- Market state
- Predicted price movement
- Portfolio position
- Volatility level
- Order imbalance

The policies of the hybrid agents constantly evolve with rewards obtained and enable the agent to dynamically adapt to changing market conditions.

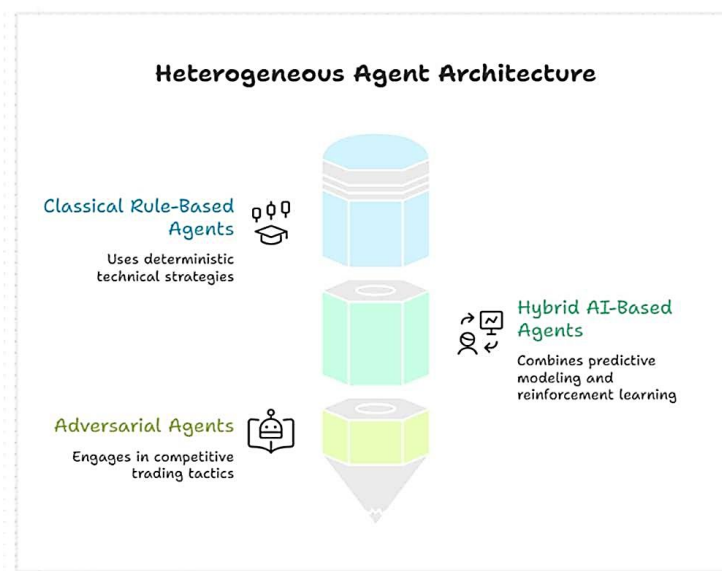


Fig. 4. Heterogeneous Agent Architecture

c. Risk-Aware Adaptation and Learning.

One important problem with reinforcement learning in finance is the risk-taking behaviour [9], [20]. Unstable trading strategies are frequently caused by pure profit maximization.

To overcome this, the reward structure will include:

- Favorable compensation of gains within portfolio.
- Penalty for large drawdowns
- Punishment on exposure to high volatility.

Risk-sensitive learning is in line with the literature of portfolio optimization and enhances long-term stability [16], [20].

Using volatility sensitive punishment, hybrid agents become trained to shun over leverage in very volatile market conditions. This design is directly related to systemic resilience to adversarial stress.

d. Adversarial Stress Modeling.

Adversarial agents are added into the simulation in order to analyze the market robustness. Such disruptive behaviors emulated by these agents include:

- Sudden large buy/sell bursts
- Artificial demand spikes
- Interim withdrawal of liquidity.
- Volatility amplification

These actions are similar to real life manipulative tactics and poisonous flow of orders in high frequency markets [12], [13], [19].

- In adversarial modeling it is possible to analyze:
- Shock propagation dynamics
- Liquidity collapse
- Flash-crash-like phenomena
- Recovery rate and adaptability.

The framework is measured by different intensities of adversarial agents in response to destabilizing events by classical and hybrid agents.

e. Stability in the Market Assessment.

In order to measure systemic resilience, a Market Stability Index (MSI) is proposed.

- Return volatility
- Maximum drawdown
- Liquidity depth
- Tail-risk indicators

These indicators are in line with empirical analyses on financial stability and systemic risk

[4], [12].

The analysis of comparative stability is done in four configurations:

1. Classical-only market
2. Hybrid-only market
3. Hybrid market (mixed classical).
4. Mixed market and antagonistic agents.

This systematic comparison enables consideration of the impact of AI-based competition on the general market strength

4. DESIGN AND IMPLEMENTATION

The set of agents trading mechanism suggested is executed as a modular simulation platform that is intended to solve the behavior of heterogeneous traders and the conflicts, which are paid by the market. The architecture is developed following the principles of agent-based computational finance, where the behaviour of the global market is a product of decentralized agent interaction instead of centralized equilibrium mechanism [5], [14].

A. System Architecture

The system has five basic components:

1. Market Engine
2. Agent Management Module
3. Hybrid Learning Module
4. Adversarial Injection Module.
5. Data mapping and Analysis Layer.



Fig. 5. Agent-Based Trading Simulation Architecture

The modular design allows grazing and directed experimentation between variants of agent compositions and adversarial strengths.

B. Market Engine

The Market Engine solves the price formation by aggregating sell and buy decisions. At each timestep, it:

- Detects the actions of all agents
- Calculates aggregate demand and supply
- One possibility is to update asset price accordingly

- Adjusts the volatility and liquidity metrics

The mechanism of the price update in line with microstructure-driven modeling approaches used in the algorithmic trading research [1], [12]. Transaction costs and liquidity constraints are added so that there are no unrealistic arbitrage conditions.

In order to ensure fairness, the execution of all the agent actions is conducted simultaneously at each timestep, so as to avoid order priority bias.

c. Agent Management Module

The Agent Management Module provides the coordination of heterogeneous agents and maintains:

- Portfolio balances
- Position sizes
- Trade history

Initial capital, strategy type and risk sensitivity are parameters that can be configured before the start of the agents. This configuration flexibility allows for comparative analysis in classical only, hybrid only, mixed and adversarial markets.

d. Hybrid Learning Module

Hybrid agents combine predictive and adaptive agents.

1. LSTM-Based Prediction

LSTM network is a processing model that works on the historical sequences of prices in order to predict the short-term direction. LSTM architectures have been well-known for modeling dependence of past data points in financial time series [17], [18]. The prediction output is considered as an input feature instead of a direct ordering command for trading, which means that forecasting and decision logic can be separated.

2. Automated Learning Decision Engine

The decision making part is a Deep Q-Network architecture [8]. The input state includes:

- Market indicators
- Volatility levels
- LSTM forecast signal
- Portfolio position
- Liquidity information

The output of DQN are the action values to Buy, Sell or Hold.

The learning stability is enhanced with:

- Experience replay
- Target network updates
- Epsilon-greedy exploration

These mechanisms are consistent with known models for reinforcement learning [9], [16].

In order to conquer the excessive risk-taking behaviour, the punishment system includes punishment of drawdowns and exposure to volatility as follows: this risk-sensitive learning by reinforcement. principles [20].

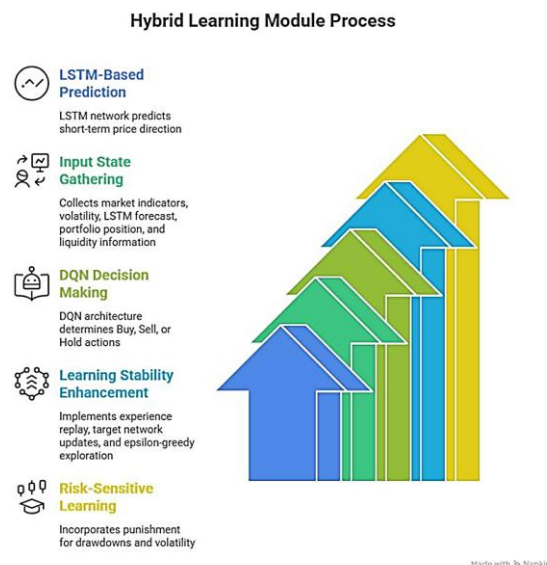


Fig. 6. Hybrid Learning Module Process

E. Injection Module Adversarial

In order to test the resilience of a system at its systemic level, adversary agents are embedded in the simulation.

These agents controlled disruptions include:

Sharp demand or supply outbursts

Liquidity withdrawal

Synchronized hypervolatility

Such behaviors imitate toxic order flow and manipulative behaviors in high frequency markets [12], [13], [19]. Adversarial intensity is parameterized, allowing the low, moderate, and extreme disruption scenarios of stress testing.

F. Analytics and Monitoring Layer

A real-time monitoring module is used for keeping track of:

- Price trajectory
- Volatility clustering
- Liquidity fluctuations
- Agent profit distribution
- Behavioral synchronization

There is a maintenance of synchronization between hybrid agents especially in check since correlated trading behaviour may be a source of increased systemic instability [11].

The results of the predictions are documented in a series of trials as part of the experiment to have some statistical reliability and reproducibility.

G. Implementation Details

The architecture is in Python using PyTorch for training of the neural network. Data preprocessing is necessary: normalization, sliding window segmentation and volatility estimation. There is hybrid acceleration using the GPU agents, while classical agents have little computational overhead.

Its modular nature also allows it to be extended to areas of multi-asset trading and into

real-world data.

5. RESULT AND DISCUSSION

The multi-agent simulation of the trading was experimented in different ways market settings in order to be able to examine the profitability, systemic stability and emergent behavior. Fourth, there were experiments in four major conditions:

1. classical rule-based agents
2. hybrid AI agents
3. mixed classical and hybrid agents
4. mixed agents.

competing in hostile market situations. Every experiment was carried out in a series of simulation experiments in order to achieve statistical reliability.

A. Performance of Hybrid and Classical Agents.

The findings show that hybrid agents always perform better than classical rule-based agents regarding the portfolio returns and flexibility to the evolving markets. Classical agents that used deterministic trading strategies like moving averages and momentum indicators had consistent and comparatively low performance. They highly relied on the historical price signals in their decisions, which curtailed their response to volatile market regimes that evolve very fast.

On the other hand, hybrid agents proved to be much more lucrative since they were able to integrate predictive modeling and adaptive decision-making. The LSTM prediction model was used to learn the time-dependent prices and the reinforcement learning module was used to train the trading policy with the help of market feedback. This has been determined in previous studies by reinforcement learning models that have shown superiority in the trading performance as compared to the conventional trading strategies [7], [16].

Nevertheless, the experiments further indicated that hybrid similar trading policies with also converged towards the agents time. This trend of convergence of behaviors resulted in the synchronised trading activities which created the market prone to market volatility. There is evidence of such co-ordination of algorithmic agents as a potential financial systemic risk in automated financial markets [11].

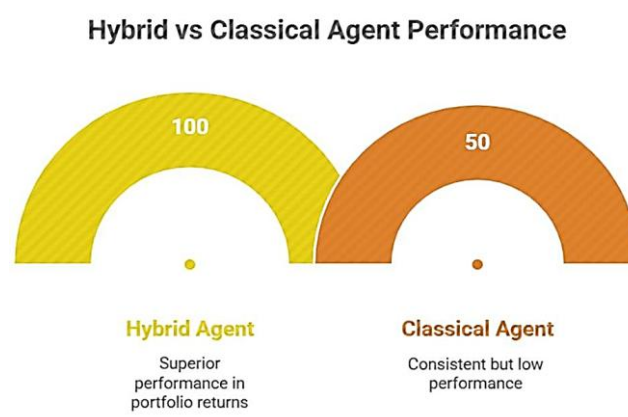


Fig. 7. Hybrid vs Classical Agent Performance

B. Effect of Adversarial Agents.

To add adversarial agents to the market simulation so. that systemic resilience could be

assessed. The disruptive behaviors were generated by agents, i.e. demand spikes and temporary liquidity withdrawal. As the example of the shows presented above indicates findings, the adversarial exercise helped to make a significant contribution increase in the volatility of the market and led to the rapid price movement.

Adversarial agents within a couple of experimental procedures generated flash-crash-like behaviours with prices falling dramatically and slowly rebounding. These findings are in concurrence with the actual market perturbations in the real world because of high frequency interactions of trading and toxic order flow [12], [13].

The agents that were hybrids reacted better to adversarial shocks than the classical agents. They were, however, quick to act at times, increasing the volatility of prices as a result of correlated trading behavior. This observation refers to one of the primary trade-offs between systematic stability and flexibility of AI-based financial markets.

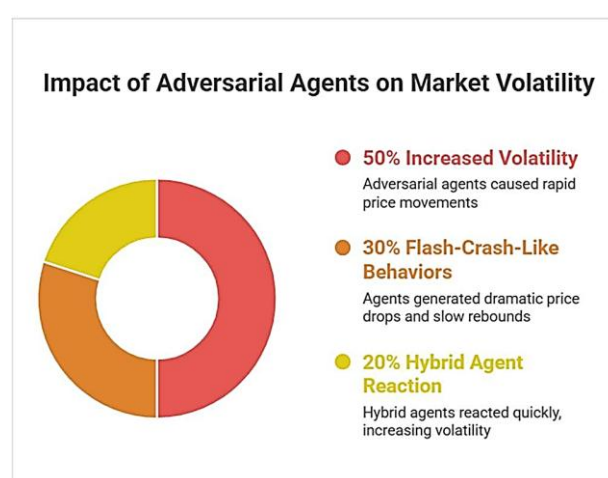


Fig. 8. Impact of Adversarial Agents on Market Volatility

c. Stability Enhancement by means of Risk-Aware Learning.

To deal with too much volatility amplification, the reinforcement learning reward scheme was changed to add penalty of drawdown and volatility exposure penalties. As risk-sensitive reward was introduced shaping, hybrid agents are now trading in a more conservative fashion way.

Empirical evidence suggests that, in case of hybrid agents that and in an experimental market are risk-sensitive, the volatility the liquidity stabilization and in the market are decreased and enhanced respectively. These results go in line with the earlier researches indicating the importance of risk-conscious financial decision-making systems reinforcement learning [16], [20].

6. CONCLUSION

This paper was about a multi agent trading simulator model aimed at examining the competitive and adversarial relationship in algorithmic financial markets. In the proposed system, classical rule-based agents, hybrid artificial Adversarial agents, and intelligence-controlled agents are linked in a single sim world. The Emergent analysis is conducted in a systematic way using framework market behavior, performance of trading and the stability of a system modeling interaction between heterogeneous

demand and agent formation of price.

The experimental works have demonstrated that hybrid agents that combine deep learning prediction and reinforcement learning decision-making are more lucrative and adjustable than traditional rule-based strategies to trade. Predictive powers of LSTM module allow identifying short-term price trends, whereas reinforcement learning enables the agents to keep the trade policies streamlined in line with the market feedback. The hybrid agents are sensitive to such properties. Varying market conditions as opposed to the classical strategies which are based on the previous indicators only.

Nevertheless, the research also has significant systemic consequences of trading AI and when a number of hybrid agents are in a shared environment, the process of their learning will be drawn towards a similar policy of making decisions. This synchronous action can have the potential to heighten the price fluctuations in the short-run and market sensitivity to shocks. This is demonstrated in the experiments where these effects are increased in situations of adversity where disruptive agents place artificial spikes of demand or liquidity withdrawal.

The recommended framework has attempted to remedy this issue considered a risk-averse policy of reinforcement learning that punishes overexposure to volatility and overdraws of a portfolio. The experimental evidence explains that these risk sensitive rewarding structures stabilize the markets and has no adverse effect on the competitive trading performance. These results underscore the need to incorporate the concepts of risk management in AI-based trading.

The framework will be continued in the future research, but with some changes realistic order book model and multi-asset trading environment, and there is a high number of agents. Also, more novel reinforcement learning algorithms, such as policy-gradient and multi-agent reinforcement learning techniques, might also augment the flexibility and steadiness of competitive trading settings.

REFERENCES

- [1] A. Cartea, S. Jaimungal, and J. Penalva, *Algorithmic and High-Frequency Trading*, Cambridge Univ. Press, 2015.
- [2] T. Hendershott et al., "Does Algorithmic Trading Improve Liquidity?" *J. Finance*, 2011.
- [3] A. Kirilenko et al., "The Flash Crash," *J. Finance*, 2017.
- [4] R. Cont, "Empirical Properties of Asset Returns," *Quant. Finance*, 2001.
- [5] L. Tesfatsion, "Agent-Based Computational Economics," 2002.
- [6] B. LeBaron, "Agent-Based Computational Finance," 2006.
- [7] J. Moody and M. Saffell, "Learning to Trade via Direct Reinforcement," *IEEE TNN*, 2001.
- [8] V. Mnih et al., "Human-Level Control Through Deep RL," *Nature*, 2015.
- [9] Y. Deng et al., "Deep RL for Portfolio Management," 2016.
- [10] Z. Jiang et al., "Deep RL Framework for Financial Trading," 2017.
- [11] C. Hommes, "Heterogeneous Agent Models," 2006.
- [12] D. Easley et al., "Flow Toxicity," *Rev. Financial Studies*, 2012.
- [13] M. O'Hara, *High-Frequency Market Microstructure*, 2015.
- [14] J. D. Farmer and D. Foley, "The Economy Needs Agent-Based Modeling," *Nature*, 2009.

- [15] W. B. Arthur et al., "Asset Pricing Under Endogenous Expectations," 1997.
- [16] Y. Nevmyvaka et al., "RL for Optimized Trade Execution," ICML, 2006.
- [17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, 1997.
- [18] T. Fischer and C. Krauss, "Deep Learning with LSTM," *EJOR*, 2018.
- [19] J.-P. Bouchaud, "Trades, Quotes and Prices," 2018.
- [20] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, no. 3, p. 032309, Sep. 2018.