A Transformer Architecture for Learning Trading Strategies

Anonymous Author(s)
Affiliation
Address
email

Unpublished working draft. Not for distribution.

Abstract

Agent-based financial market simulations have advanced significantly through the integration of deep reinforcement learning (dRL), enabling adaptive trading agents such as the previously introduced trained response order network (TRON). Although TRON agents demonstrated improved market performance compared to traditional heuristic methods, their reliance on recurrent neural networks limits their ability to capture complex, long-range dependencies inherent in sequential financial data. In this paper, we introduce TRONformer, a novel trading agent architecture that combines transformer-based self-attention mechanisms with policy-gradient dRL. Our approach leverages the superior representational power of transformers to effectively model intricate market dynamics and dependencies. Empirical evaluations across multiple simulated financial market scenarios demonstrate that TRONformer agents consistently outperform previous TRON agents in profitability.

1 Introduction

Agent-based financial market simulations have proven indispensable for understanding the complex interactions among strategic traders, market mechanisms, and economic outcomes. Traditional approaches in market simulations, such as the Zero Intelligence (ZI) agents introduced by Gode and Sunder [3], have highlighted how minimal strategic sophistication can still produce efficient market equilibria. More recently, ***deep reinforcement learning*** (deep RL, or dRL) [17] techniques have enabled the development of more sophisticated adaptive strategies, leading to notable performance improvements in simulated environments.

Prior work introduced the ***Trained Response Order Network*** (TRON) agent [9], which leverages deep Q learning [11] methods and recurrent neural networks to adaptively condition trading strategies on observed market states. TRON agents demonstrated improvements over simpler heuristic strategies by dynamically optimizing their order submission behavior, effectively balancing immediate trade opportunities with expected long-term returns.

However, despite TRON's promising results, its recurrent network structure may not fully capture complex long-term dependencies within sequential market data. Transformers [18], with their self-attention mechanisms, have shown superior capabilities in modeling intricate sequential dynamics across diverse domains such as natural language processing [1] and time-series forecasting [24]. Recognizing this potential, we propose the ***TRONformer agent***, an approach to learning trading strategies based on a transformer architecture. TRONformer combines the powerful representational capabilities of transformers with the adaptive capacity of dRL.

We evaluate TRONformer by training a trading agent within an agent-based simulation environment. The particular market scenarios are adapted from past work on RL-derived strategies, enabling us to calibrate the observed improvements in performance relative to reported baselines.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2 Related Work

2.1 Agent-Based Simulation of Financial Markets

Agent-based models (ABMs) have long been used [7, 12] to study financial markets by modeling the behaviors and interactions of heterogeneous traders within a simulated exchange. ABMs provide a bottom-up perspective, where macro-level market outcomes emerge from the micro-level actions of agents. This framework enables direct investigation of how specific agent strategies, information structures, or institutional designs influence market-level behavior. ABMs are particularly well-suited for exploring strategic adaptation, coordination failures, and market anomalies that are difficult to capture analytically.

Empirical game-theoretic analysis (EGTA) [20] is a methodology for game-theoretic reasoning based on ABMs. The key idea is to induce a game model (the ***empirical game***) from simulation data. In practice, EGTA starts by defining a finite set of strategies, typically based on parametrized heuristics, and generating data by simulating profiles of these strategies playing the game. By solving the resulting empirical game, the analyst identifies equilibria within this restricted strategy set. Iterative versions of EGTA may introduce additional strategies based on intermediate analyses. For instance, PSRO [6] employs dRL to derive new strategies that optimally respond to a solution of the current empirical game. Numerous works have applied EGTA to financial market scenarios [8, 10, 13–15, 19, 21].

2.2 Transformers

Transformers [18] are a type of machine learning model that has become the standard way of handling sequential data, such as sentences in language or time series in finance. Traditional models like recurrent neural networks (RNNs) process information step by step, which makes it hard for them to keep track of events that happened far in the past. Transformers solve this problem by letting every piece of information in the sequence directly look at, and weigh, every other piece of information through a mechanism called *self-attention*.

We are motivated to consider transformers as a representation for trading strategies because market observations exhibit both short-horizon signals (e.g., order-flow imbalance) and long-horizon signals (e.g., trends in fundamental values). LSTM models compress history into a single hidden state, risking loss of long-range information. Self-attention circumvents this limitation, letting the agent selectively focus on the subset of past events most relevant for a particular trading decisions.

3 Market Environment

We conduct all experiments in PyMarketSim [10], a discrete–event agent-based financial market simulator that instantiates a single-asset continuous limit order book (LOB). PyMarketSim is designed to support training strategies with dRL, employing efficient order-book data structures [23] for fast execution and providing wrappers for interface to dRL libraries.

3.1 Market and Valuation Model

Our market model follows prior work in agent-based finance [4, 19]. Agent valuations are comprised of two components: a per-unit ***fundamental value*** common to all agents, and a ***private valuation*** over the quantity of holdings specific to each agent. The fundamental value of an asset is generated by a mean-reverting stochastic process f_t, characterized by parameters including a long-run mean ($\bar{f}$), a mean reversion coefficient ($\kappa \in [0, 1]$), and shock variance ($\sigma_s^2 \in [0, \infty)$). Agents observe the current fundamental f_t on arrival at t, and forecast the terminal fundamental value $\tilde{f}_T$ as a weighted average of f_t and the long-run mean $\bar{f}$

Private valuations are based on the model of Goettler et al. [4] and are parameterized by the maximum holding $q_{\max} \in [0, \infty)$, and the private value variance $\sigma_{pv}^2 \in [0, \infty)$. Each agent receives a vector Θ_i of private values, with size $|\Theta_i| = 2q_{\max}$, representing marginal values for incremental trades. Components θ_i^k are drawn from $\mathcal{N}(0, \sigma_{pv}^2)$ and sorted in descending order:

$$\Theta_i = (\theta_i^{-q_{\max}+1}, \ldots, \theta_i^0, \theta_i^1, \ldots, \theta_i^{q_{\max}}).$$

Unpublished working draft. Not for distribution.

An agent's total portfolio value, v_T, comprises cash (c_T), holdings' market value, and cumulative private valuations:

$$v_T = c_T + q_T f_T + \begin{cases} \sum_{j=1}^{q_T} \theta_i^j & q_T > 0 \\ 0 & q_T = 0 \\ \sum_{j=q_T+1}^{0} -\theta_i^j & q_T < 0 \end{cases} \tag{1}$$

3.2 Zero Intelligence Agents

Zero Intelligence (ZI) agents [3], as the name suggests, implement a basic trading strategy that does not entail any informed analysis of the market situation. The ZI strategy simply submits orders requesting a particular surplus, randomly chosen, relative to their current valuation. The current valuation can be calculated using $\tilde{f}_T$ from (**??**), and the private value vector θ, with q indicating current holdings. The requested surplus s is drawn from a uniform distribution, $s \sim \mathcal{U}(R_{\min}, R_{\max})$, thus determining the ZI limit order price p:

$$p = \tilde{f}_T + \begin{cases} \theta^{q+1} - s & \text{if assigned buy,} \\ \theta^q + s & \text{if assigned sell.} \end{cases} \tag{2}$$

Our ZI agents further incorporate a threshold parameter $\eta \in [0, 1]$ to assess the attractiveness of immediate execution at the prevailing market price p^* (BID for selling, ASK for buying).

3.3 TRON Agent

The ***Trained Response Order Network*** (TRON) agent [10] extends the ZI framework by incorporating deep reinforcement learning. TRON agents employ neural networks trained with policy-gradient methods to adaptively adjust order parameters (s, η) in response to observed market states. This adaptability allows agents to balance immediate trading opportunities with expected long-term returns. The original TRON implementation used an LSTM backbone [5] to capture sequential dependencies.

TRON defines rewards in terms of portfolio value changes between consecutive actions. Specifically, the reward earned from the most recent action at time t_l is computed as the difference in portfolio value since that action:

$$r_{t_l} = v_t - v_{t_l},$$

where v_i is defined in Equation 1.

To provide a denser reward signal and accelerate learning, we introduce a learned reward model that predicts whether an order will execute before the agent's next action. This model is trained on equilibrium data generated from ZI agents. During training, the model augments the reward as follows:

$$sr_{t_l} = (1 - \gamma) r_{t_l} + \gamma\, p(x) s(x),$$

where $p(x)$ is the model's predicted probability that the order executes, and $s(x)$ is the surplus demanded by the agent.

4 TRONformer

Motivation. TRON's recurrent core compresses the entire observation history into a single hidden state. While effective for short horizons, this mechanism scales poorly as the informational size of order-book signals grows. We therefore replace the LSTM with an *encoder-only [2] transformer* whose self-attention layers give every timestep direct, content-addressable access to every other timestep, eliminating the path-length bottleneck that impedes long-range credit assignment.

State tokenization. At each decision epoch the agent observes the information it had at each previous point it entered the market. Each event $\mathbf{x}_\ell \in \mathbb{R}^{d_{\text{raw}}}$ is first linearly embedded and then followed by addition of rotary positional encodings [16]. A special `[CLS]` token is prepended to serve as a global context vector in the spirit of BERT [2].

The TRONFormer architecture is depicted in Figure 1.

Unpublished working draft. Not for distribution.

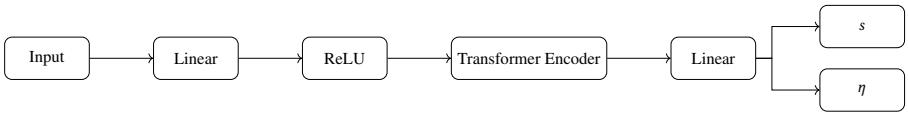

Figure 1: Architecture of the TRONformer agent. The network derives order parameters (s, η) from its observations of market states. It is trained with profit-based reward signals, shaped using both portfolio value changes and a learned execution probability model.

5 Experiments

Our experiments use PyMarketSim [9] and focus on a single market environment. Each run spans $t = 2000$ trading periods with an arrival rate of $\lambda = 0.012$ and mean reversion coefficient $\kappa = 0.01$. The variance of the fundamental value is set to $\sigma_s^2 = 2.0 \times 10^4$, while the variance of private values is $\sigma_p^2 = 2.0 \times 10^7$. We simulate $n = 16$ agents with 15 equilibrated ZI agents and one TRONformer agent.

5.1 ZI Regret Study

Our goal is to measure the gain from adaptivity attributable to RL policies, and to do so we need a background market that is consistent, not an arbitrary hand-tuned mix of heuristic agents. We use methods from EGTA to solve for a symmetric Nash equilibrium across ZI agents. In the resulting ZI profile, no ZI agent can profitably deviate against the population. Using this equilibrium as the background removes confounds due to mis-specification of ZI parameters and isolates the value added by adaptivity. An equilibrium baseline was also used as a basis for comparison in Wright and Wellman [22]. Solving the payoff matrix yields a symmetric equilibrium at $[460, 540, 0.5]$ in the environment.

5.2 TRONformer Performance

Against the equilibrium background of 15 ZI agents, the TRONformer agent consistently achieved higher utilities than both ZI and the original TRON. The equilibrium ZI baseline reached an average utility of about 1262, while TRON improved on this by roughly 8% with a best run of 1359. TRONformer delivered the strongest performance, reaching 1479, corresponding to a $\sim 17\%$ improvement over ZI and a clear margin over TRON. These results demonstrate that replacing the recurrent backbone with a transformer encoder yields measurable gains in profitability, validating the role of self-attention in capturing long-range order-book dependencies.

5.3 Ablations

To assess design choices, we ran two ablations. First, disabling the learned reward model ($\gamma = 0$) reduced utility to 1435, indicating its role in stabilizing early training. Second, replacing deep RL with the tabular Q-learning agent of Wright and Wellman [21] yielded 1278, underscoring the advantage of function approximation in this setting.

6 Conclusions

In this paper, we introduced TRONformer, a trading agent architecture that integrates transformer-based self-attention mechanisms with deep reinforcement learning. This approach specifically addresses the limitations inherent in previous recurrent neural network-based methods, such as the TRON agent, by effectively modeling complex, long-range dependencies characteristic of sequential financial market data.

Overall, our findings underscore the effectiveness of transformer-based architectures in capturing intricate market dynamics and adapting strategies based on historical trading signals. Future directions for research include extending the TRONformer framework to multi-agent settings, and exploring additional transformer modifications tailored specifically for financial and multiagent data.

Unpublished working draft. Not for distribution.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. doi:10.48550/arXiv.2005.14165 arXiv:2005.14165 [cs].

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. doi:10.48550/arXiv.1810.04805 arXiv:1810.04805 [cs].

[3] Dhananjay K. Gode and Shyam Sunder. 1993. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy* 101, 1 (1993), 119–137. https://www.jstor.org/stable/2138676 Publisher: University of Chicago Press.

[4] Ronald L. Goettler, Christine A. Parlour, and Uday Rajan. 2009. Informed traders and limit order markets. *Journal of Financial Economics* 93 (2009), 67–87.

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (Nov. 1997), 1735–1780.

[6] Marc Lanctot, Vinicius Zambaldi, Audrūnas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *31st Annual Conference on Neural Information Processing Systems (NeurIPS)* (Long Beach, CA). 4190–4203.

[7] Blake LeBaron. 2000. Agent-Based Computational Finance: Suggested Readings and Early Research. 24 (2000), 679–702.

[8] Buhong Liu, Maria Polukarov, Carmine Ventre, Lingbo Li, Leslie Kanthan, Fan Wu, and Michail Basios. 2022. The spoofing resistance of frequent call markets. In *21st International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 825–832.

[9] Chris Mascioli, Anri Gu, Yongzhao Wang, Mithun Chakraborty, and Michael Wellman. 2024. A financial market simulation environment for trading agents using deep reinforcement learning. In *Proceedings of the 5th ACM International Conference on AI in Finance*. ACM, Brooklyn NY USA, 117–125. doi:10.1145/3677052.3698639

[10] Chris Mascioli, Anri Gu, Yongzhao Wang, Mithun Chakraborty, and Michael P. Wellman. 2024. A financial market simulation environment for trading agents using deep reinforcement learning. In *5th International Conference on Artificial Intelligence in Finance (ICAIF)* (Brooklyn). 117–125.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. doi:10.48550/arXiv.1312.5602 arXiv:1312.5602 [cs].

[12] R. G. Palmer, W. Brian Arthur, John H. Holland, Blake LeBaron, and Paul Tayler. 1994. Artificial economic life: A simple model of a stockmarket. *Physica D: Nonlinear Phenomena* 75 (1994), 264–274.

[13] Ji Qi and Carmine Ventre. 2022. Accounting for strategic response in limit order book dynamics. In *24th International Conference on Principles and Practice of Multi-Agent Systems* (Valencia). 630–639.

[14] Sriram Bharadwaj Rangarajan and Carmine Ventre. 2025. Impact of pinging in financial markets: An agent based study. In *17th International Conference on Agents and Artificial Intelligence (ICAART)*. 172–183.

[15] L. Julian Schvartzman and Michael P. Wellman. 2009. Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In *8th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)* (Budapest). 249–256.

[16] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing* 568 (Feb. 2024), 127063. doi:10.1016/j.neucom.2023.127063

[17] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: an introduction* (second edition ed.). The MIT Press, Cambridge, Massachusetts.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. http://arxiv.org/abs/1706.03762 arXiv:1706.03762 [cs].

[19] Elaine Wah, Mason Wright, and Michael P. Wellman. 2017. Welfare Effects of Market Making in Continuous Double Auctions. *Journal of Artificial Intelligence Research* 59 (2017), 613–650.

[20] Michael P. Wellman, Karl Tuyls, and Amy Greenwald. 2025. Empirical Game-Theoretic Analysis: A Survey. *Journal of Artificial Intelligence Research* 82 (2025), 1017–1076.

[21] Mason Wright and Michael P. Wellman. 2018. Evaluating the stability of non-adaptive trading in continuous double auctions. In *17th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)* (Stockholm). 614–622.

[22] Mason Wright and Michael P Wellman. 2018. Evaluating the Stability of Non-Adaptive Trading in Continuous Double Auctions. (2018).

Unpublished working draft. Not for distribution.

[23] Peter R Wurman, William E Walsh, and Michael P Wellman. 1998. Flexible double auctions for electronic commerce: theory and implementation. *Decision Support Systems* 24, 1 (Nov. 1998), 17–27. doi:10.1016/S0167-9236(98)00060-8

[24] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: beyond efficient transformer for long sequence time-series forecasting. doi:10.48550/arXiv.2012.07436 arXiv:2012.07436 [cs].

Unpublished working draft.
Not for distribution.