Reinforcement Learning and Strategic Behavior in a Financial Tug-of-War Model

By Brendon Bazzani, Cameron Holtz, Emmanuel Dey, Lesley Machimbidza
Department of Mathematics, Dartmouth College

Abstract

This project investigates a stylized financial tug-of-war model, where two trading agents compete in a zero-sum stochastic game to steer an asset's price toward their preferred absorbing boundary (high or low) via strategic buy/sell actions. Using dynamic programming and Nash equilibrium analysis, we derive analytical benchmarks for optimal play. We then implement a reinforcement learning (RL) approach Q-learning with ε-greedy exploration where agents learn through repeated self-play or against fixed-strategy opponents. Learned policies are compared to optimal strategies derived via value iteration. Results show RL agents gradually converge to equilibrium behaviors, with early-phase deviations due to limited state exposure. Eventually, agents mirror optimal boundary-pulling strategies and achieve 95–100% payoff efficiency in most states. When facing suboptimal opponents, RL agents exploit inefficiencies, surpassing Nash baseline payoffs. We highlight implications for real-world markets, where traders adapt under bounded rationality and incomplete information, often diverging from theoretical equilibria.

Background

Introduction

Financial markets often resemble competitive games, with traders using strategies to push asset prices in favorable directions. We model such a scenario as a two-player zero-sum stochastic game in which the "state" is a discrete price level and two adversarial trading agents exert opposing influences on price movements. This tug-of-war metaphor captures how buyers and sellers compete: if bullish trades dominate, the price is pulled upward, whereas dominant selling pressure pulls it downward. Eventually, the price may hit an upper or lower bound (reflecting extreme outcomes like a price target or stop-loss), at which point one agent "wins" and the game ends. This formalization offers a clean and tractable lens through which to examine strategic price dynamics in simplified market settings. The game is discrete in both time and state, with transitions between price levels determined probabilistically by the players' simultaneous actions. These actions, which metaphorically represent net buying or selling pressure, result in random walk dynamics with a directional bias depending on whether agents act in unison or in opposition. Importantly, this setup allows us to investigate not only how rational agents *should* behave, but also how boundedly rational agents *do* behave when they must learn effective strategies through experience.

Significance

This game-theoretic setup allows us to apply dynamic programming and reinforcement learning to analyze strategic behavior. Under perfect rationality, agents solve the **Bellman equation** to determine the value function V(s), which represents the expected payoff from state s when both players act optimally. The general equation is:

$$V(s);=;\max_{a_1\in A_1};\min_{a_2\in A_2};\mathbb{E}\Big[,V(s'),\Big|,s,a_1,a_2\Big],$$

He e next state s′. This saddle-point form reflects the zero-sum nature of the game.

Furthermore, our specific model is,

$$V(s) \;=\; \tfrac{1}{2}\,V(s+1) \;+\; \tfrac{1}{2}\,V(s-1),$$

with boundary conditions

$$V(0) = 0, \quad V(N) = 1.$$

In parallel, we implement **Q-learning** for reinforcement learning agents that lack prior knowledge of the game's dynamics. By training agents through repeated play, either against fixed opponents or via self-play, we assess how closely their learned behavior aligns with the optimal strategies derived from the Bellman equation

Models and Methods

We model the financial tug-of-war as a turn-based game played in discrete time, where two competing traders influence the movement of an asset's price. One trader, representing bullish market behavior, consistently tries to push the price upward. The other, representing bearish sentiment, applies downward pressure. The price itself is simplified into discrete levels, forming a finite set of states. These states range from a minimum level (representing the lower price bound) to a maximum level (the upper bound). The game ends when the price hits either extreme, with one trader declared the winner based on which direction the price reached.

At every step of the game, both players simultaneously choose an action. These actions correspond to their trading intent: pushing the price up or pulling it down. However, the outcome is not deterministic. Instead, it is governed by probability. When both players choose to push in the same direction, either both upward or both downward, the price is likely to move in that direction, but not with certainty. The strength of the price movement reflects a bias toward the combined push. On the other hand, if the players oppose each other, one pushing up and the other down, the forces cancel out, and the price has an equal chance of moving up or down. This random element reflects real-world market uncertainty, where even coordinated actions cannot fully control outcomes.

The central goal for each trader is to steer the price toward their preferred terminal state: the bullish trader wants to reach the upper bound, and the bearish trader wants the price to fall to the lower bound. In order to determine the best possible strategy for each player, we turn to the tools of dynamic programming. We define a value function that tells us the probability that the bullish trader wins when the price is at a given level, assuming both players make optimal choices from that point forward.

Solving for this function involves reasoning recursively: at any non-terminal state, the optimal action depends on the expected outcome of all possible future paths. The bullish player aims to maximize their chance of success, while the bearish player tries to minimize it. The result is a game of opposing interests, where each player reacts not only to the current state but to how the other might respond.

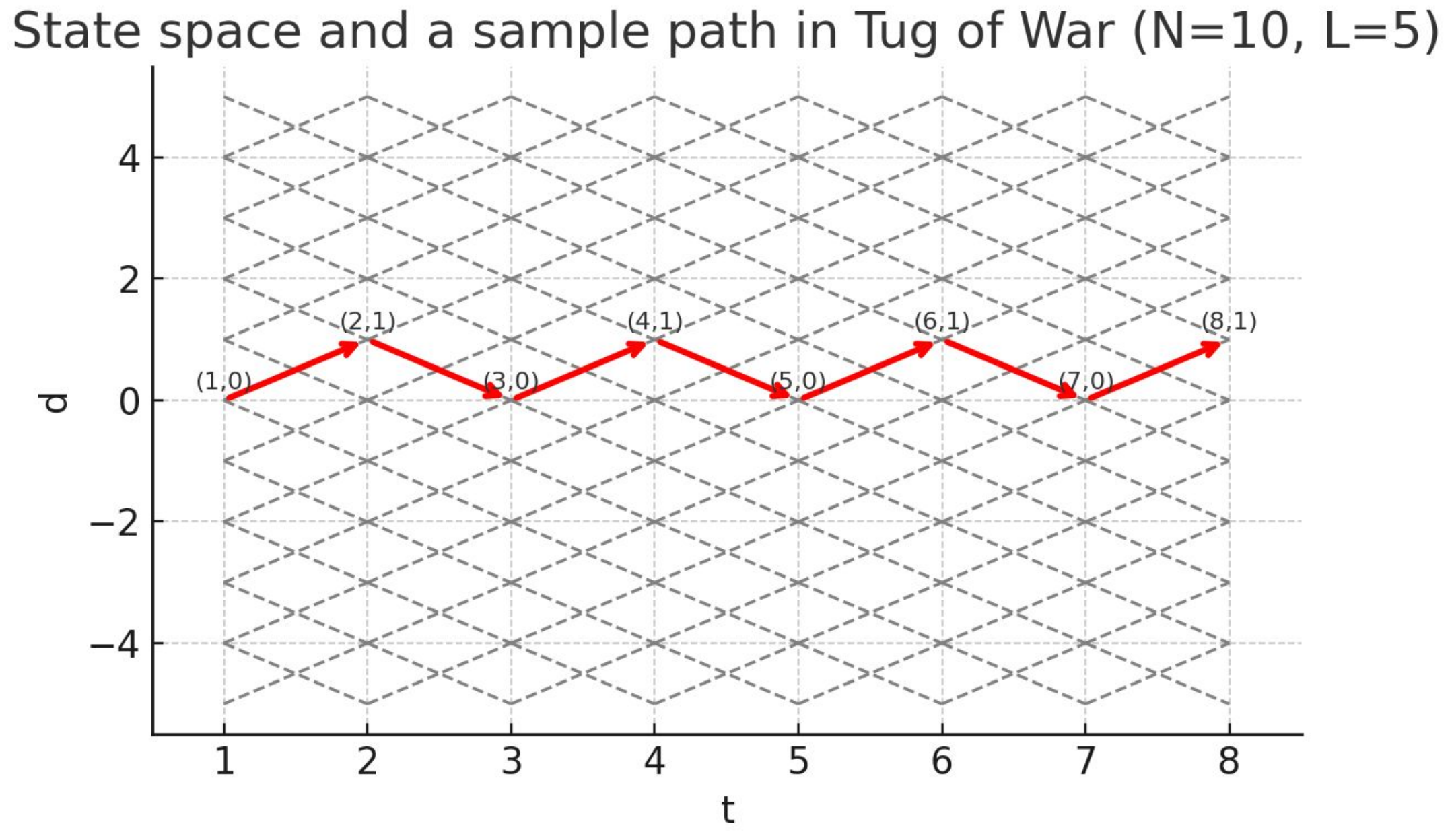
In the special case where the game is symmetric, meaning both players have the same influence and the same rules for how actions affect the price, the best strategy becomes straightforward. Each player should always apply pressure in the direction of their goal: the bullish trader always pushes upward, and the bearish trader always pushes downward. When this occurs, the game reduces to a fair random walk, where the price moves unpredictably until it reaches one of the boundaries. In this case, the probability that the bullish trader wins is simply proportional to how far the current price is from the lower bound.

To explore how real-world traders might behave when they do not have perfect information or computational power, we introduce reinforcement learning. Instead of calculating optimal moves in advance, reinforcement learning agents improve over time by learning from experience. We use a specific method called Q-learning, where each agent maintains an evolving estimate of how good each possible action is in each state. These estimates are updated after every game based on whether the agent won or lost and what actions it took along the way.

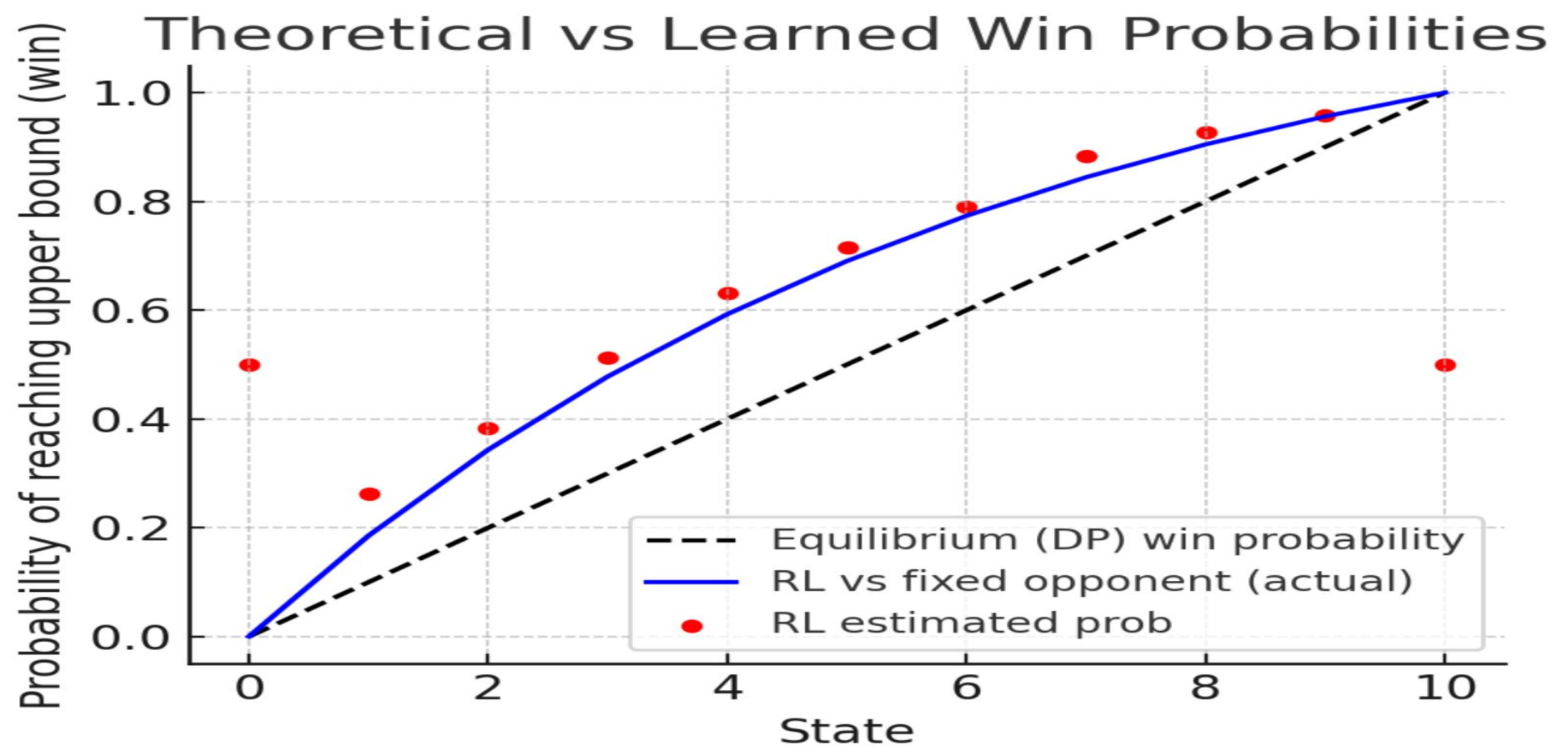
Over the course of many training episodes, the agent plays repeatedly and gradually learns which actions tend to lead to success. Sometimes the agent acts according to its current best guess; other times, it explores less certain options to improve its understanding. This exploration ensures that the agent doesn't miss out on potentially better strategies.

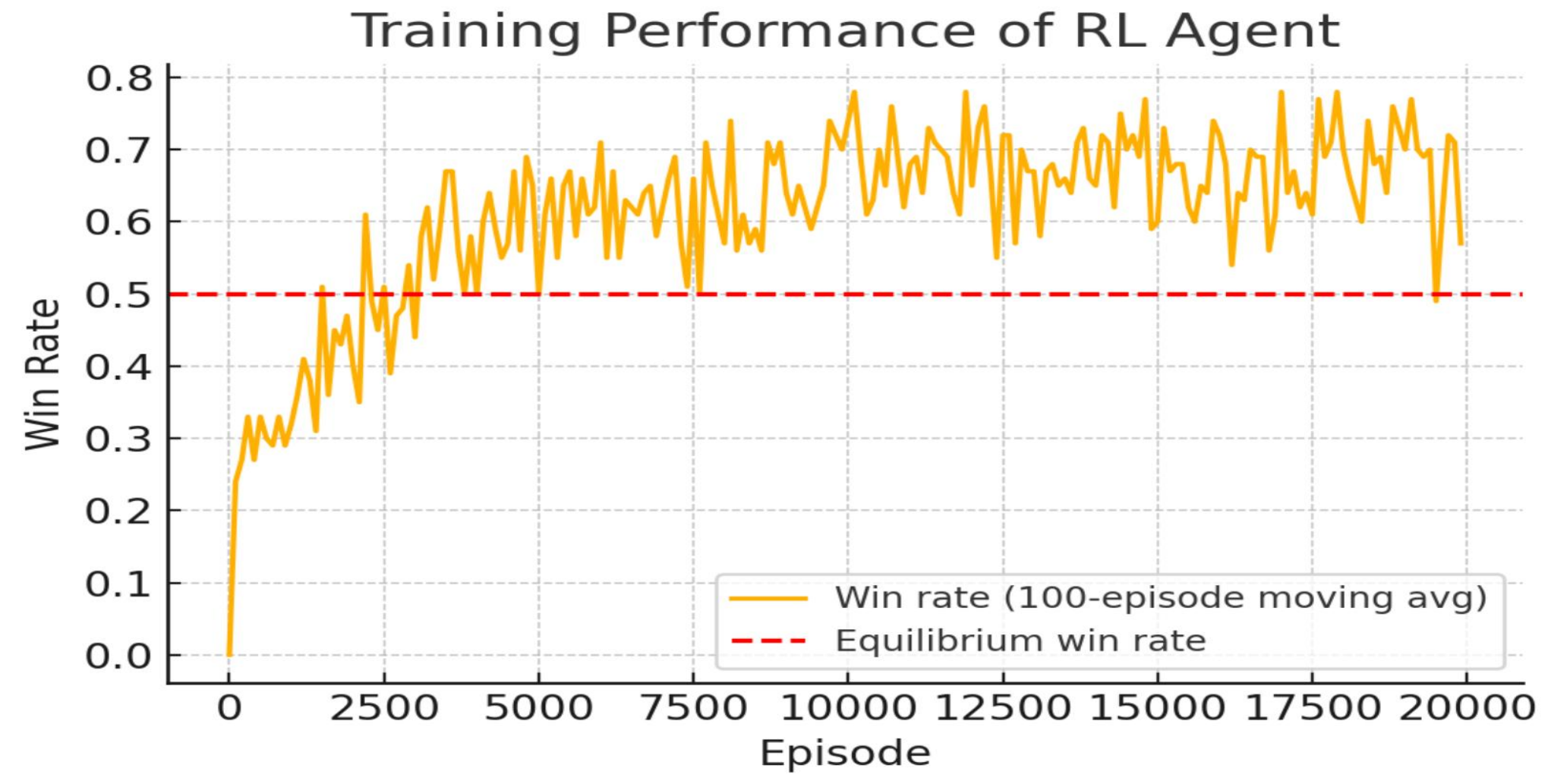
There are two main ways to train these agents. One method involves placing the learning agent against a fixed opponent that always behaves in the same way. This lets us observe how well the agent can adapt to exploit a predictable adversary. The other method is called self-play, where both agents learn at the same time, constantly adjusting to each other's strategies. This dynamic environment better reflects real market behavior, where traders evolve in response to one another.

As training continues, the reinforcement learning agents typically converge to strategies that closely resemble the theoretically optimal ones. Their win probabilities stabilize and begin to mirror the expected results derived from dynamic programming. By comparing these learned outcomes with exact solutions, we gain insight into how close real learning-based behavior can come to rational, game-theoretic ideals—and where it might diverge due to limitations in information, time, or adaptation.


State space and a sample path in Tug of War (N=10, L=5)

Results


Theoretical vs Learned Win Probabilities


Training Performance of RL Agent

Using dynamic programming, we derived the analytical benchmark for the symmetric tug-of-war model, where two agents compete to pull an asset's price toward their respective absorbing boundary. By solving the difference equations via value iteration, we obtained the equilibrium value function V(s)=s/N , which represents the probability that Player A wins from state *s* under optimal play; correspondingly, Player B wins with probability N−s/N The optimal (Nash equilibrium) policy is for Player A to always choose Up (U) and Player B to always choose Down (D) in every non-terminal state. Any deviation from this strategy such as Player A choosing D or Player B choosing U immediately favors the opponent, who can continue pulling in their direction to gain a probabilistic advantage. This dynamic results in an unbiased martingale process, where the price fluctuates until one of the absorbing boundaries is reached. While we acknowledge that asymmetric versions of the game (e.g., when stalemate transitions are biased with p≠0.5 may require mixed strategies, our analysis focuses on the symmetric case to provide a rigorous baseline for evaluating reinforcement learning (RL) agent behavior. The equilibrium value V(s) serves as a performance benchmark: any RL policy achieving a higher win probability must be exploiting a suboptimal opponent; lower values indicate that the learned policy has not fully converged to optimal play.

Discussion and Conclusion

Our study explores the interplay between theoretically optimal strategies derived via dynamic programming under perfect rationality and the adaptive behaviors of reinforcement learning (RL) agents with bounded rationality and limited information. In the symmetric tug-of-war model, optimal play requires each player to always push in their direction, yielding a linear win probability V(s)=s/N Without prior knowledge, RL agents gradually learned this strategy through self-play and adaptation. When facing suboptimal opponents, they exceeded equilibrium payoffs by exploiting mistakes, mirroring real-world scenarios where adaptive traders capitalize on inefficiencies. However, learning involved transient suboptimal behavior, particularly in rarely visited states analogous to irrational actions like panic selling or premature exits in financial markets. These inefficiencies declined over time as agents refined their value estimates, reflecting how arbitrage can self-correct mispricings. Our results suggest RL offers a powerful lens for modeling strategic behavior in adversarial financial environments where perfect information and rationality do not hold. Future work will extend the model to include partial observability, richer action spaces, transaction costs, and risk preferences to better reflect market dynamics. We also plan to scale the model using deep multi-agent RL and explore empirical applications, such as interpreting order book competition as a tug-of-war, to assess how learned strategies align with real trading behavior