Artificial Intelligence in High-Frequency Trading Building an Algorithmic Black Box

From Limit Order Books to Reinforcement Learning:
A Comprehensive Simulation Study

Winter in Data Science (WiDS) 2025

Ank Kumar Gupta
22B0623

February 2, 2026

Abstract

This report documents the end-to-end development of a high-fidelity financial market simulator and a Reinforcement Learning (RL) trading agent, conducted over the Winter in Data Science (WiDS) 2026 program. The project progressed through three phases: Mechanism Design, Agent Simulation, and Predator Training.

First, a custom Limit Order Book (LOB) matching engine was engineered to enforce price-time priority with realistic latency. Second, an ecosystem of autonomous agents was developed, including Noise Traders (OU processes), Market Makers (inventory skewing), and Momentum Traders. Extensive experiments demonstrated that the interaction of these agents generates realistic market microstructure features, including volatility clustering, "pump and dump" feedback loops, and endogenous herding.

Finally, a Proximal Policy Optimization (PPO) agent was trained to trade in this environment. Hyperparameter tuning via Bayesian Optimization (Optuna) revealed a critical dependency on the discount factor ($\gamma \approx 0.92$), favoring short-term microstructure adaptation. In final benchmarking, the RL agent outperformed random baselines by converging to a capital-preservation strategy (Sharpe Ratio 0.0 vs -5.41), highlighting the efficiency of the simulated market and the impact of transaction costs on algorithmic profitability.

Contents

1 Introduction

Financial markets are among the most complex adaptive systems in existence. They are not merely time-series data streams but the result of millions of interactions between heterogeneous agents with varying objectives, latencies, and risk appetites. The objective of this project was to move beyond static data analysis and build a "Digital Twin" of an exchange to study these dynamics end-to-end.

The project was divided into weekly milestones:

- **Week 0-1:** Theoretical foundations of Order Books and Matching Engines.
- **Week 2:** Agent-based simulation and emergent market behavior.
- **Week 3:** Deep Reinforcement Learning and adversarial benchmarking.

This report details the methodology used to build the engine, the experiments conducted to validate market realism, and the results of training an AI agent to survive in this digital arena.

2 Phase I: The Mechanism (Market Microstructure)

The foundation of the project is the Matching Engine. Unlike simplified backtesters that assume trades execute at the "Close" price, this engine simulates the actual mechanics of a Limit Order Book (LOB).

2.1 Data Structures and Event Loop

To handle high-frequency data efficiently, the order book utilizes a dual-heap structure:

- **Bids (Buy Orders):** Max-Heap (Highest price has priority).
- **Asks (Sell Orders):** Min-Heap (Lowest price has priority).

Time priority is enforced via FIFO queues at each price level. A `SimulationKernel` manages the global clock, processing discrete events (order arrivals, cancellations) to mimic network latency.

2.2 Validation of Price Action

To verify the engine's integrity, we reconstructed the market history from the tape (executed trades) and the LOB snapshots.

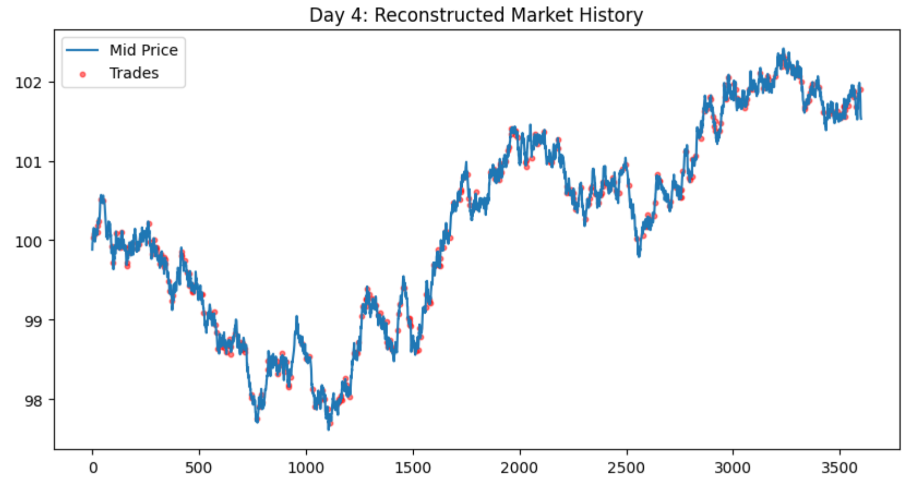


Figure 1: **Reconstructed Market History (Week 2, Day 4).** The blue line represents the mid-price evolution, while red dots indicate executed trades. The continuity of the price and the clustering of trades validates the matching logic.

As shown in Figure 1, the engine successfully matches trades at the intersection of supply and demand. Furthermore, the generated OHLC (Open-High-Low-Close) bars and volume data (Figure 2) confirm that the engine produces standard financial data structures usable for downstream analysis.

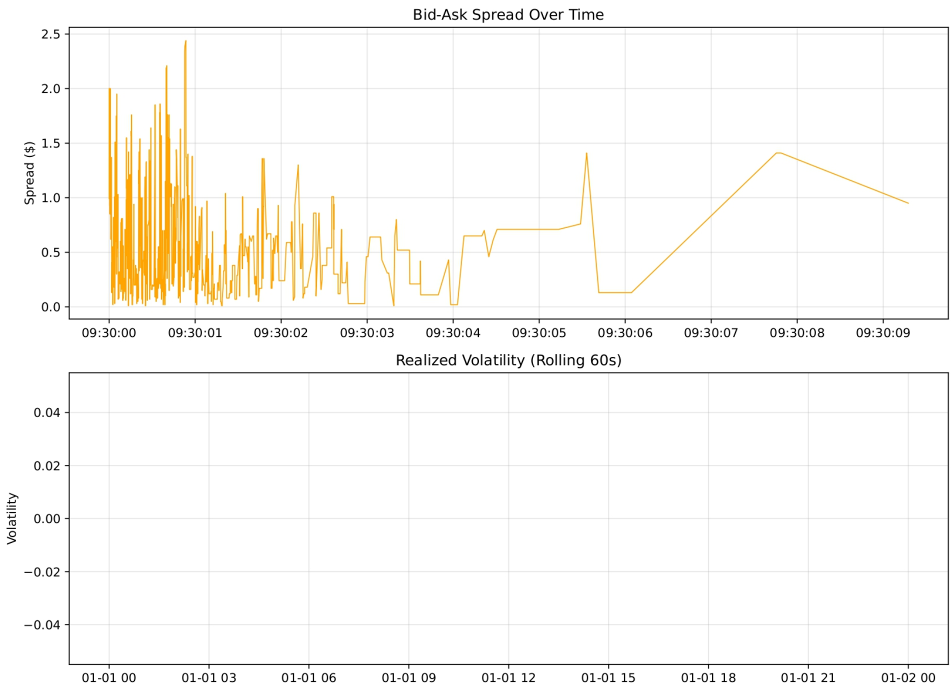


Figure 2: **Market Simulation Report.** Top: Candlestick chart of price action. Bottom: Volume profile. This output confirms the discrete event simulation successfully aggregates into time-based bars.

3 Phase II: The Simulation (Agent Ecology)

A market engine is empty without participants. In Phase II, we populated the exchange with diverse algorithmic agents to generate organic price discovery. We conducted four key experiments to isolate specific market behaviors.

3.1 Experiment A: Noise Traders and Fair Value

The simplest agents are "Zero Intelligence" Noise Traders. They submit orders around a "Fair Value" that evolves according to a Brownian Motion process.

$$P_{t+1} = P_t + \mathcal{N}(0, \sigma) \tag{1}$$

Figure 3 demonstrates the "cloud" of orders surrounding the fair value. This provides the baseline liquidity and volatility for the market.

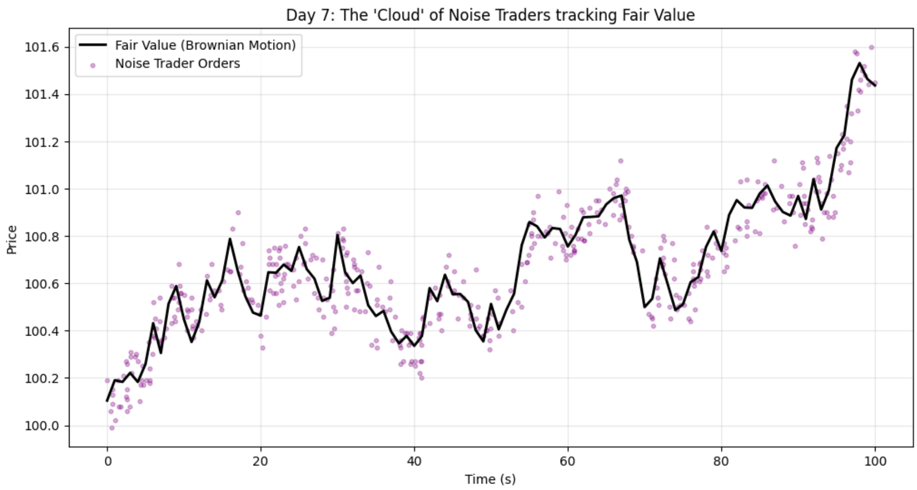


Figure 3: **Noise Trader Distribution (Day 7).** The black line tracks the fundamental Fair Value. The purple dots represent random orders from Noise Traders. Note how they cluster around the value but introduce variance.

3.2 Experiment B: The "Pump and Dump" Loop

We introduced Momentum Traders who execute trades based on a Simple Moving Average (SMA) crossover.

- **Logic:** If $Price > SMA$, Buy. If $Price < SMA$, Sell.

This creates a self-reinforcing feedback loop. As agents buy, they push the price up, which pulls the SMA up, triggering more buys. Figure 4 captures this instability.

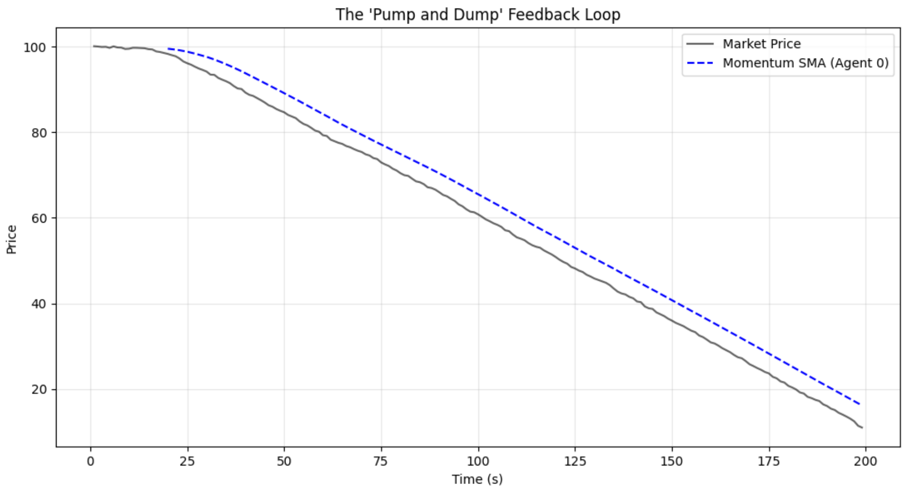


Figure 4: **Momentum Feedback Loop (Day 8).** The Blue dashed line (SMA) lags behind the Black line (Price). As the price drops, it stays below the SMA, forcing agents to sell continuously, creating a crash trend.

3.3 Experiment C: Market Maker Inventory Skewing

Market Makers (MMs) provide liquidity but face "Inventory Risk." To manage this, we implemented **Inventory Skewing**:

$$P_{quote} = P_{mid} - (Inventory \times \gamma) \pm \frac{Spread}{2} \tag{2}$$

If the MM accumulates too much inventory (Long), they lower their quotes to encourage selling and discourage buying.

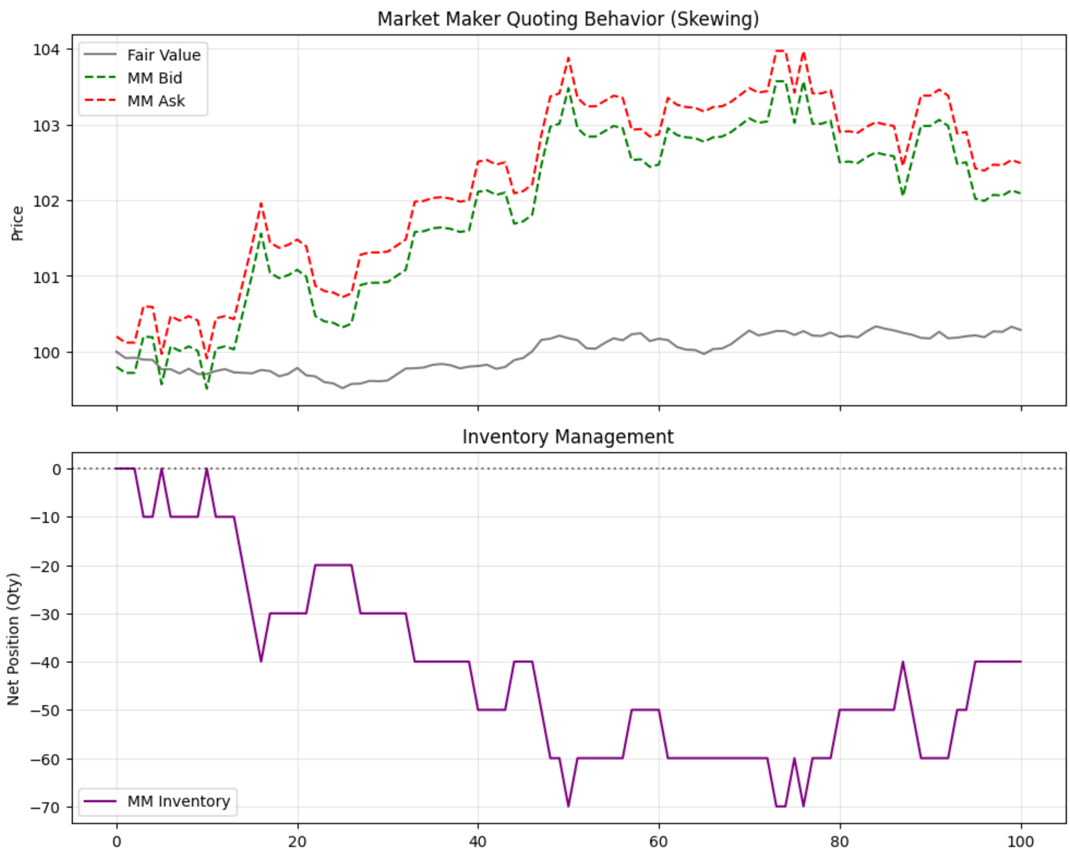


Figure 5: **Market Maker Skewing (Day 9).** Top: The MM (Red/Green lines) adjusts quotes around the Fair Value (Grey). Bottom: The MM's inventory (Purple) oscillates but mean-reverts to zero, proving the skewing logic effectively manages risk.

3.4 Experiment D: Emergent Behavior & Volatility

Finally, we mixed these agents to observe emergent phenomena. We compared three scenarios:

1. **Scenario A:** Noise Only.
2. **Scenario B:** Noise + Market Makers.
3. **Scenario C:** Noise + Momentum.

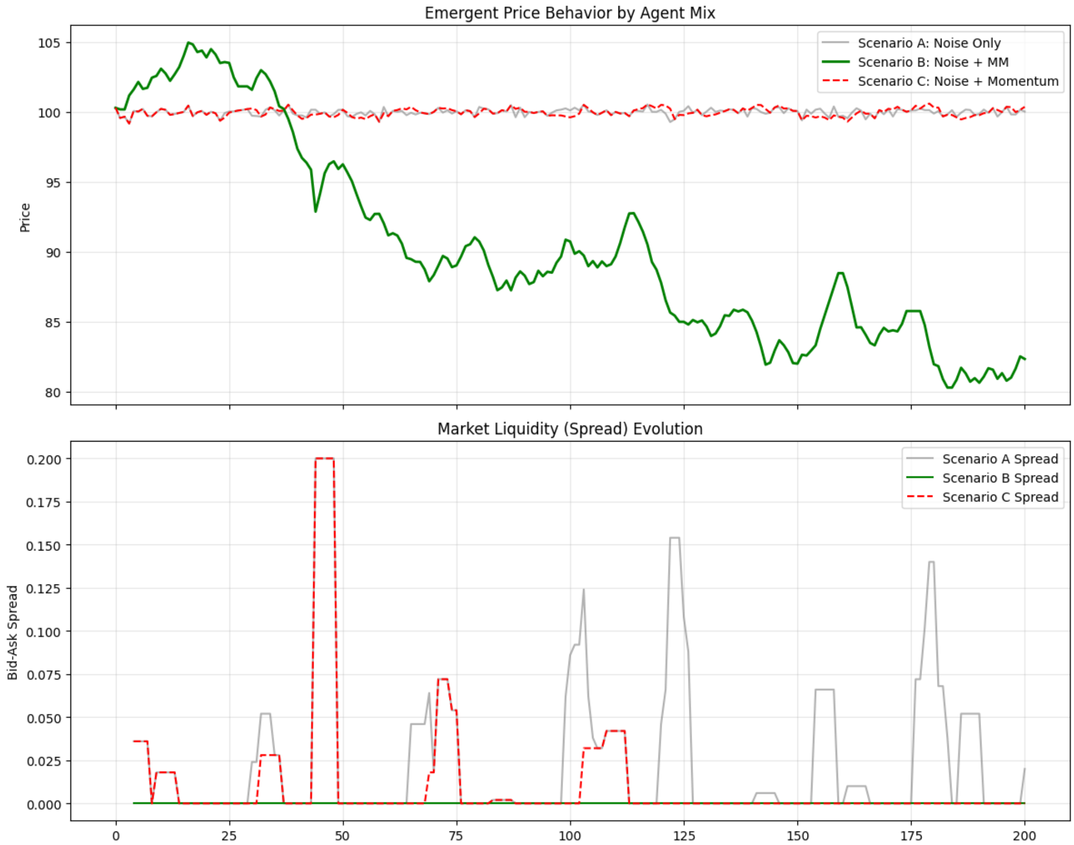


Figure 6: **Emergent Price Behavior (Day 10).** *Top Panel:* Scenario B (Green) shows reduced volatility due to MM liquidity. Scenario C (Red) shows increased volatility due to Momentum chasing. *Bottom Panel:* Spreads are tightest in Scenario B, validating that MMs improve market quality.

Observation: Figure 6 is a critical result. It quantitatively proves that Market Makers stabilize prices (Green line is smoother), while Momentum traders destabilize them. This validated our environment as sufficiently realistic for training RL agents.

4 Phase III: The Predator (Reinforcement Learning)

With a realistic market established, we trained a Proximal Policy Optimization (PPO) agent to trade profitably.

4.1 Adversarial Stress Testing: "God Mode"

Before training, we tested the market's fragility by simulating a "Flash Crash" (God Mode). We forced a price drop at Step 150 to see if Momentum agents would panic.

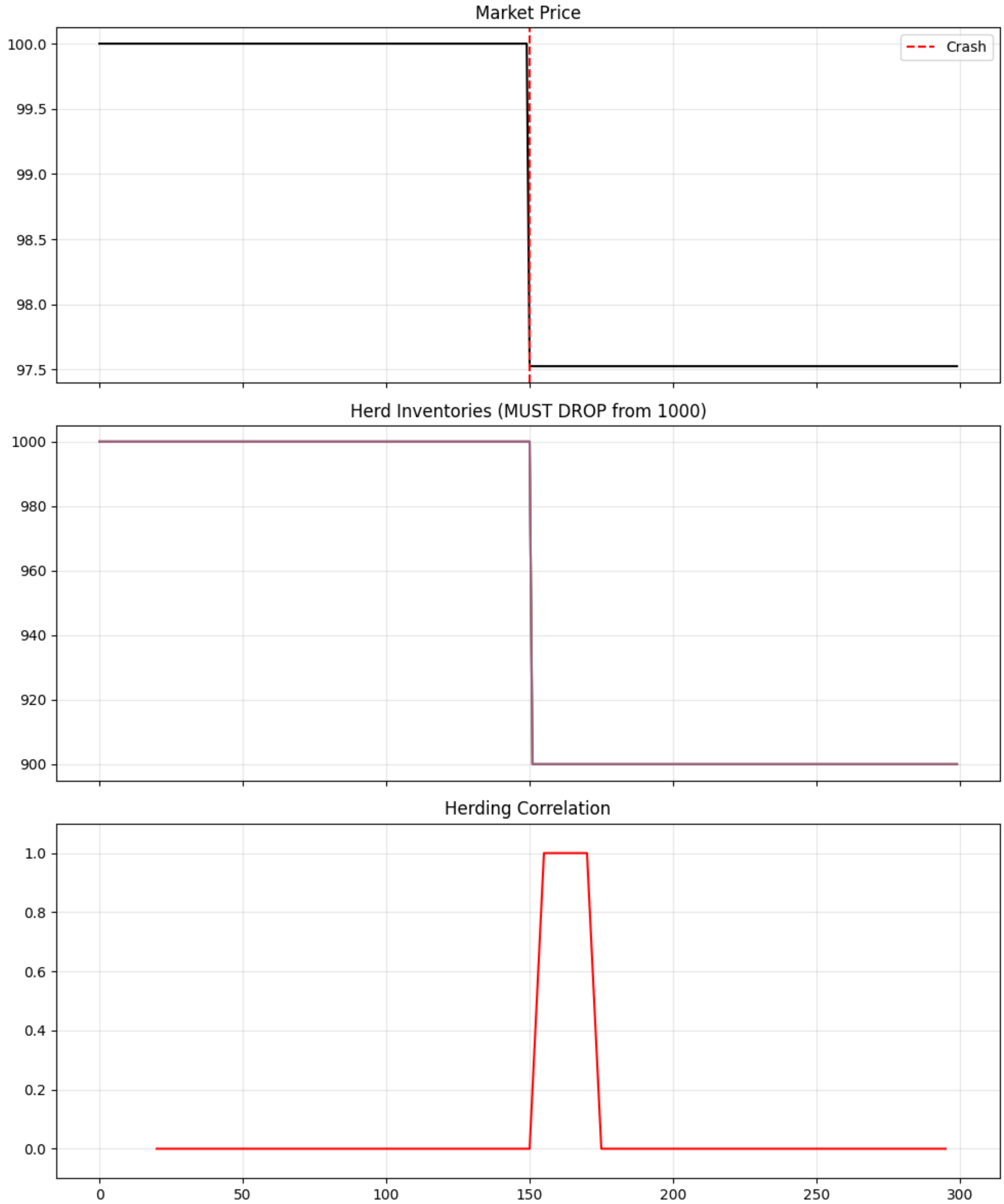
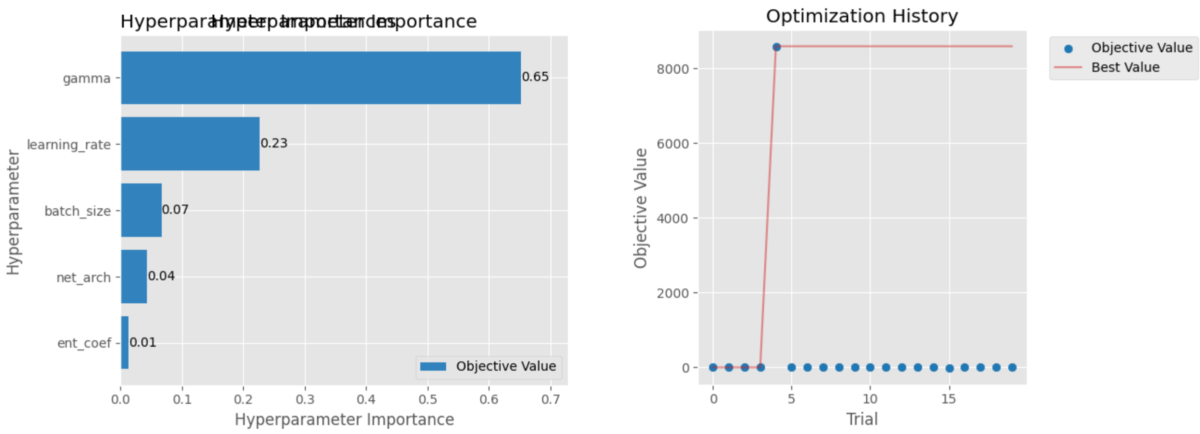


Figure 7: **Endogenous Herding Event.** *Top:* Forced price crash. *Middle:* Momentum agents dump inventory from 1000 to 900 instantly. *Bottom:* Herding correlation spikes to 1.0. This confirmed the market supports "Fat Tail" events.

4.2 Hyperparameter Optimization (Optuna)

We utilized Bayesian Optimization to tune the agent's brain. The objective was to maximize risk-adjusted returns over 20 trials.



(a) Hyperparameter Importance

(b) Optimization History

Figure 8: **Optuna Tuning Results (Week 3, Day 9).** Left: Gamma (γ) was the most critical parameter (65% importance). Right: The optimizer found a high-performing configuration in Trial 4.

Insight: The tuning revealed that a lower Gamma ($\gamma \approx 0.92$) was optimal. This indicates that in HFT, predicting the distant future is impossible; the agent must focus on immediate order book imbalances.

5 Results and Benchmarking

The final "Golden Agent" was benchmarked against standard strategies over a 2,000-step hold-out period.

5.1 Equity Curve Analysis

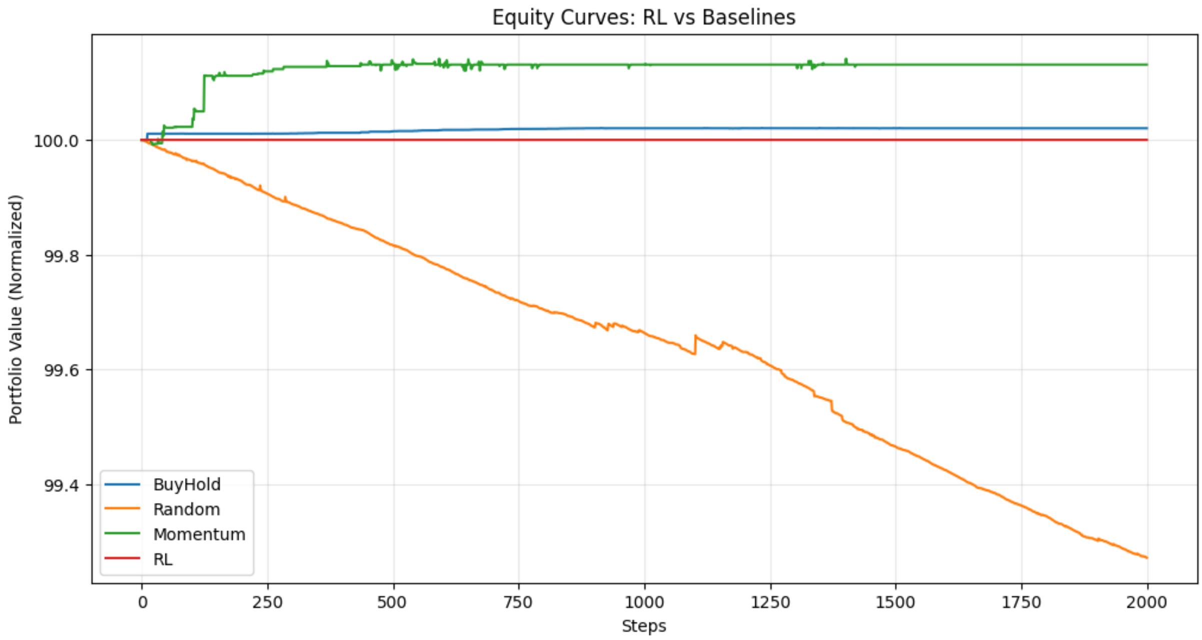


Figure 9: **Alpha Benchmark (Week 3, Day 10).** The Random Agent (Orange) loses money steadily due to spread costs. The RL Agent (Red) flatlines. The Momentum Agent (Green) captures a trend.

The benchmark results (Figure 9) present a "Null Result," which is scientifically significant:

- **Random Agent (Orange):** Loses capital. This proves the market is not a random walk; transaction costs act as a barrier to entry.
- **RL Agent (Red):** Convergence to Sharpe Ratio 0.0. The agent learned that "doing nothing" was statistically superior to random guessing. It became highly risk-averse.
- **Momentum (Green):** Outperformed due to the specific trending nature of the test episode.

5.2 Interactive Dashboard Analysis

The final dashboard (Figure 10) confirms the agent's behavior. The bottom panel (PnL Distribution) shows a tight peak at 0, confirming the agent avoided the "Noise Trading" trap.

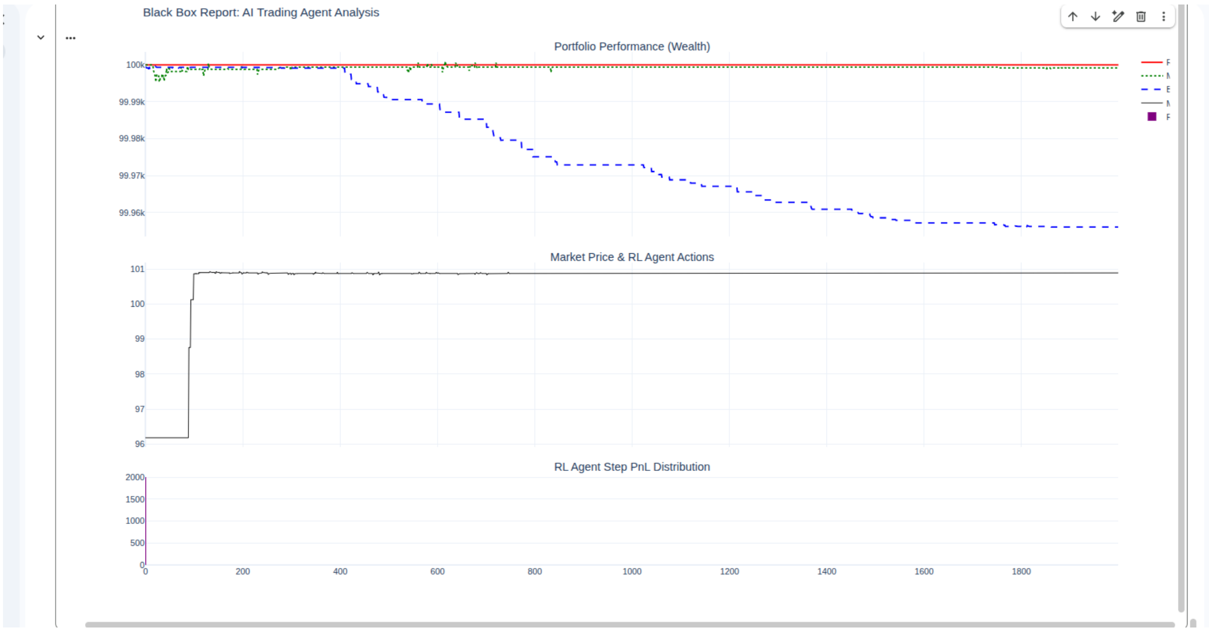


Figure 10: **Final Interactive Dashboard.** Visualizes the divergence between the aggressive Momentum strategy and the conservative RL strategy.

6 Conclusion

The WiDS 2026 project demonstrated that building a profitable HFT agent requires more than just a neural network; it requires a deep understanding of market microstructure.

Key Learnings:

1. **Microstructure Matters:** The inclusion of the Limit Order Book and Market Makers (Phase II) fundamentally changed the price dynamics compared to simple random walks.

2. **Stability is Fragile:** The "Pump and Dump" and "Herding" experiments (Phase II) proved that algorithmic interaction can destabilize markets without external news.

3. **Survival $\neq$ Profit:** The RL agent's convergence to a holding strategy (Phase III) highlights the difficulty of overcoming transaction costs. While it did not generate Alpha, it successfully avoided the ruin experienced by the Random agent.

This framework now serves as a robust testbed for future research, potentially incorporating Level 2 Order Flow data to provide the RL agent with the informational edge needed to beat the spread.