(Deep) Learning to Trade: An Experimental Analysis of AI Trading and Market Outcomes

Ivan Gufler[1] Francesco Sangiorgi[2] Emanuele Tarantino[3]

[1]Luiss [2]Frankfurt School of Finance & Management [3]Luiss

Bristol University, March 5, 2025

Motivation & Research Question

- Financial markets exhibit return predictability arising from:
  - Public signals (e.g., fundamentals, firm characteristics).
  - Latent demand shocks (e.g., investor sentiment, large trades).
- Reinforcement learning (RL) traders do not assume a known structure–they learn from *experience*.

Research Questions:

- Can AI-driven investors detect and exploit return predictability?
- How do they influence market efficiency, liquidity, and price formation?

Goal: Understand how AI strategies learn from and reshape asset prices.

Algorithmic Behavioral Finance

- Behavioral finance focuses on human biases.
- As AI-based trading grows, the decision-makers are algorithms.
- Goldstein, Spatt, and Ye (2021): *"Just as insights into human behavior from the psychology literature spawned the field of behavioral finance, so can insights into algorithmic behavior (or the psychology of machines) spawn an analogous blossoming of research in algorithmic behavioral finance."*

What We Do

- **Combine a demand-based asset pricing approach** (Koijen and Yogo, 2019) with **deep reinforcement learning (DRL)**:
  - Prices reflect both fundamentals and latent demand from a representative investor.
  - AI traders learn endogenously and have price impact.
- **Compare AI outcomes to a rational expectations (RE) benchmark**:
  - Do AI traders discover predictable signals?
  - Do AI traders internalize their price impact and that of competing AIs?
  - How does AI trading shape market efficiency and liquidity?

Preview of Results

- **Qualitative Alignment with Rational Benchmark:**
  - AI portfolios largely mirror the comparative statics from the RE case.
  - Agents learn to decode price information and internalize price impact.
- **Negative Learning Externality with Many AI Traders:**
  - Quantitatively, AI traders deviate from RE as competition increases.
  - Multiple AIs generate additional noise, degrading each other's learning.
  - This distortion leads to only partial internalization of price impact, degrading performance relative to the RE benchmark.
- **Market Efficiency and Liquidity:**
  - AI trading increases market efficiency making returns less predictable.
    - But not as much as RE traders would.
  - AI traders provide less liquidity compared to RE traders.
    - AIs do not react to unexpected changes in the environment as RE would.

Literature

- **AI & Market Outcomes:** Biais et al. (2015); Brogaard et al. (2015); Foucault et al. (2016)
- **Reinforcement Learning in Finance and Economics:** Calvano et al. (2020); Colliard et al. (2023); Abada and Lambin (2023); Johnson et al. (2023); Dou et al. (2023); Yang (2024).

  Our paper differs in (i) focus (predictability vs. coordination/collusion), (ii) approach (embedding RL in empirically plausible settings), and (iii) methods (Deep RL with continuous action spaces vs. tabular-Q learning).
- **Behavioral Biases in Model-Free Algorithms:** Barberis and Jin (2023)
- **Demand-Based Asset Pricing:** Koijen and Yogo (2019)

Roadmap

1. Market Environment
2. Rational Expectation Benchmark
3. Empirical Design
4. DRL Implementation
5. Experiments
6. Conclusion

Market Environment: Setup

- Time: $t = 0, 1, 2, \ldots$
- N risky assets with supply S_n (exogenous) and price $P_{n,t}$ (endogenous)
  - Risky assets pay dividends $D_{n,t}$; we assume $D_{n,t}/P_{n,t-1}$ is exogenous.
- A riskless asset with exogenous gross return R_f.
- Two types of agents:
  - **AI Traders** ($j = 1, \ldots, J$), holding $S^j_{n,t}$ shares of asset n at time t.
  - **Representative Investor** (marginal investor, or "the market"), holding the residual supply $S_n - \sum_{j=1}^{J} S^j_{n,t}$.

Representative Investor's Demand

Log-exponential form as in (Koijen and Yogo (2019)):

$$\frac{w_{n,t}}{w_{0,t}+\gamma_t} = \exp\left(\beta_0(p_{n,t}+s_n) + \sum_{k=1}^{K-1} \beta_k\, x_{k,n,t} + \beta_K + \epsilon_{n,t}\right).$$

- $w_{n,t}, w_{0,t}$: portfolio weights in risky vs. risk-free.
- γ_t: fraction of assets consumed by the representative investor.
- $p_{n,t} = \log(P_{n,t})$, $s_n = \log(S_n)$.
- $x_{k,n,t}$: firm characteristics (e.g., book-to-market, profitability...).
- $\epsilon_{n,t}$: *latent demand*.
- $\{\beta_0, \beta_1, \cdots\}$ shape the demand elasticity of the representative investor.

Equilibrium Prices

- Solve for $\{P_{n,t}\}$ taking $\{S^j_{n,t}\}$ as given and addition assumptions.
- We obtain the following **equilibrium price**

$$p_{n,t} = -s_n + \frac{-\ln\left(1-\alpha^{AI}_{n,t}\right) + \sum_{k=1}^{K-1} \beta_k x_{k,n,t} + \beta_K + \epsilon_{n,t} + \ln(D_{M,t}) + \phi}{1-\beta_0}.$$

  - $\alpha^{AI}_{n,t} = \frac{\sum_{j=1}^{J} S^j_{n,t}}{S_n}$: fraction of supply demanded by AIs.
  - $D_{M,t} = \sum_{n=1}^{N} S_n D_{n,t}$: aggregate dividend (growing at rate g).
  - ϕ: constant capturing g, R_f, λ (consumption rate).

Equilibrium Prices

- Solve for $\{P_{n,t}\}$ taking $\{S^j_{n,t}\}$ as given and addition assumptions.
- We obtain the following **equilibrium price**

$$p_{n,t} = -s_n + \frac{-\ln\left(1-\alpha^{AI}_{n,t}\right) + \sum_{k=1}^{K-1} \beta_k x_{k,n,t} + \beta_K + \epsilon_{n,t} + \ln(D_{M,t}) + \phi}{1-\beta_0}.$$

  - $\alpha^{AI}_{n,t} = \frac{\sum_{j=1}^{J} S^j_{n,t}}{S_n}$: fraction of supply demanded by AIs.
  - $D_{M,t} = \sum_{n=1}^{N} S_n D_{n,t}$: aggregate dividend (growing at rate g).
  - ϕ: constant capturing g, R_f, λ (consumption rate).
- Larger $\alpha^{AI}_{n,t} \implies$ higher $p_{n,t}$; price impact depends on β_0.
- β_0 closer to 1 means demand is less elastic $\implies$ stronger price impact.

Learning & Return Predictability

- Assume $\alpha^{AI}_{t+1} = 0$. In equilibrium, the (log) capital gain equals

$$p_{n,t+1} - p_{n,t} = \frac{\ln\left(1 - \alpha^{AI}_{n,t}\right) + \sum_{k=1}^{K-1} \beta_k \Delta x_{k,n,t+1} + \Delta \epsilon_{n,t+1} + \log(1+g)}{1 - \beta_0}$$

- Stock characteristics and latent demand follow an AR(1) process:

$$x_{k,n,t+1} = c_{k,n} + \rho_{k,n} x_{k,n,t} + \eta_{k,n,t+1},$$
$$\epsilon_{n,t+1} = c_{\epsilon,n} + \rho_{\epsilon,n} \epsilon_{n,t} + \xi_{n,t+1}.$$

Learning & Return Predictability

- Assume $\alpha^{AI}_{t+1} = 0$. In equilibrium, the (log) capital gain equals

$$p_{n,t+1} - p_{n,t} = \frac{\ln\left(1 - \alpha^{AI}_{n,t}\right) + \sum_{k=1}^{K-1} \beta_k \Delta x_{k,n,t+1} + \Delta\epsilon_{n,t+1} + \log(1+g)}{1 - \beta_0}$$

- Stock characteristics and latent demand follow an AR(1) process:

$$x_{k,n,t+1} = c_{k,n} + \rho_{k,n} x_{k,n,t} + \eta_{k,n,t+1},$$
$$\epsilon_{n,t+1} = c_{\epsilon,n} + \rho_{\epsilon,n}\epsilon_{n,t} + \xi_{n,t+1}.$$

Return predictability arises from mean reversion in $\{x_{k,n,t}\}, \epsilon_{n,t}$.

- Stock characteristics are public information, the latent demand is not. However...

Decoding Prices

- The price function can be written as

$$p_{n,t} = f_n(\text{public information}) + \frac{\epsilon_{n,t}}{1-\beta_0}. \tag{1}$$

- Investors with rational expectations (RE) know the price function in (1).
- For these agents, $p_{n,t}$ fully reveals $\epsilon_{n,t}$.
- AI investors have no knowledge of the data generating process.
- Can they learn it?

One-Period RE Benchmark: Setup

RE benchmark: useful to evaluate the AIs' portfolio policies, both qualitatively and quantitatively.

- J risk-neutral, rational, strategic speculators enter at t, exit at $t+1$.
- Trade in risky asset n and the riskless asset.
- Initial wealth equal to a share ω of the mkt. cap. of asset n.
- Each chooses portfolio share $\theta^j_{n,t} \in [0,1]$ in the risky asset.
- Next period, $\theta^j_{n,t+1} = 0$ (liquidation).

One-Period RE Benchmark: Information

- Before trading they all share the same information

$$\mathcal{I}_t = \{me^*_{n,t}, \{x_{n,k,t}\}_{k=1}^{K-1}\}.$$

where we define "adjusted market equity" as

$$me^*_{n,t} = p^*_{n,t} + s_n - \frac{\log(D_{M,t+1})}{1-\beta_0},$$

and $p^*_{n,t}$ is the pre-trade price

$$p^*_{n,t} = -s_n + \frac{\sum_{k=1}^{K-1}\beta_k x_{k,n,t} + \beta_K + \epsilon_{n,t} + \ln(D_{M,t}) + \phi}{1-\beta_0}.$$

- $\mathcal{I}_t$ fully reveals $\epsilon_{n,t}$:

$$E(\epsilon_{n,t}|\mathcal{I}_t) = (1-\beta_0)me^*_{n,t} - \left(\sum_{k=1}^{K-1}\beta_k x_{k,n,t} + \beta_K + \phi\right) = \epsilon_{n,t}.$$

REE: Portfolio Choice

A Rational Expectation Equilibrium (REE) is a set of portfolio shares $\{\theta^j_{n,t}\}$ that are individually optimal for each speculator given their information and given the price formation process.

Proposition

An equilibrium exists, is unique, and is symmetric. In equilibrium:

(i) θ *is decreasing in adjusted market equity* $me^*_{n,t}$;

(ii) θ *is increasing in asset characteristic* $x_{n,k,t}$ *iff* $\beta_k(\rho_{n,k} - \rho_{\epsilon,n}) > 0$;

(iii) θ *is decreasing in* ω *(size effect).*

(iv) *Fixing* ωJ, θ *is increasing in* J *(competition effect).*

REE: Market Efficiency

- The data generating process for returns is

$$R^e_{n,t+1} = g^*(\mathcal{I}_t) + e_{n,t+1},$$

where $e_{n,t+1}$ is exogenous, and $g^*(\cdot)$ depends on speculators' trading.

- **Market Efficiency**: the fraction of unpredictable return variation

$$\mathcal{ME} = \frac{Var(e_{n,t+1})}{Var\left[g^*(\mathcal{I}_t)\right] + Var(e_{n,t+1})}.$$

Proposition

Market efficiency increases with the size of each speculator, ω, and, given the total size of speculators ωJ, with the number of speculators, J.

REE: Market Liquidity

- Consider a temporary supply shock $s_n \to s_n + \sigma$.
  - In the case without speculators, $\frac{\partial p_{n,t}}{\partial \sigma} = -1$
  - The price drop leads to a reversal in the next period.
  - Anticipating this, speculators buy the asset, mitigating the price impact of the shock.
- **Liquidity**:

$$\mathcal{L} = 1 + E\left(\left|\frac{\partial p_{n,t}}{\partial \sigma}\right|_{\sigma=0}\right).$$

Proposition

Liquidity increases with the size of each speculator, ω, and, given the total size of speculators ωJ, with the number of speculators, J.

Algorithmic Learning vs. Rational Benchmark

- **Rational:** Perfectly extracts $\epsilon_{n,t}$ from prices, trades on mispricing internalizing price impact.
- **AI (RL):**
  - Learns the structure from data from experience, step by step.
  - May not fully internalize price impact.
  - May not remove predictable components as much as the RE benchmark.
- Compare how closely AI mimics the RE outcomes in terms of:
  - Portfolio policy and trading profits.
  - Market outcomes: efficiency and liquidity

Demand Estimation: Koijen & Yogo (2019)

Objective: Estimate how investors' portfolio weights respond to stock characteristics and price, while allowing for a latent demand component.

- Following Koijen and Yogo (2019), for investor i and quarter t:

$$\frac{w_{i,n,t}}{w_{i,0,t}} = \exp\left(\beta_{i,t}^{me}\,\mathrm{me}_{n,t} + \beta_{i,t}^{be}\,\mathrm{be}_{n,t} + \cdots + \beta_{i,t}^{mkt}\,\mathrm{mkt}_{n,t} + \beta_{i,t}^{0} + \epsilon_{i,n,t}\right).$$

- Characteristics: market equity, book equity, investment growth, dividend/book, profitability, market beta.
- $\epsilon_{i,n,t}$: investor-specific latent demand.
- Data sources:
  - SEC 13F filings (investor holdings).
  - CRSP/Compustat for firm characteristics (1982:Q2 – 2021:Q4).

Representative Investor Demand calibration

- Aggregate investor-level demand coefficients into a **representative investor**:

$$\beta_k = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{\text{AUM}_{i,t}}{\sum_{i=1}^{I} \text{AUM}_{i,t}} \beta_{i,t}^{k}.$$

- Similarly, we compute the representative investor's latent demand as

$$\epsilon_{n,t} = \sum_{i=1}^{I} \frac{\text{AUM}_{i,t}}{\sum_{i=1}^{I} \text{AUM}_{i,t}} \epsilon_{i,n,t}.$$

.

Simulation of Characteristics & Dividends

- For each stock n, we fit AR(1) processes for characteristic $(\text{be}_n, \text{prof}_n, \text{inv}_n, \text{div}_n, \text{mkt}_n)$ and for the latent demand $\epsilon_{n,t}$.
- Dividends:
  - Sample dividend yield from data.
  - $D_{n,t+1} = (\text{div. yield}) \times P_{n,t}$.
- Calibrate λ and g to match average observed returns.
- Set R_f equal to the mean over the sample period.
- Use estimated processes and parameters to simulate equilibrium prices.

Deep Deterministic Policy Gradient (DDPG)

- **DDPG** is a reinforcement learning algorithm for **continuous action spaces**.
- Standard Q-learning deals with discrete actions; DDPG uses:
  - **Actor network**: selects actions (portfolio weights).
  - **Critic network**: evaluates the quality of these actions.
- The agent learns an optimal trading policy to maximize cumulative rewards (portfolio returns).

How DDPG Learns Over Time

Step-by-step:

1. Observe market state s_t (prices, characteristics, holdings).
2. Actor selects action $a_t = \mu_{\theta^\mu}(s_t)$ (portfolio weights).
3. Environment returns reward r_t and next state s_{t+1}.
4. Critic updates value function:

$$y = r_t + \gamma\, Q_{\theta^{Q'}}\big(s_{t+1}, \mu_{\theta^{\mu'}}(s_{t+1})\big).$$

5. Actor improves its policy via policy gradient:

$$\nabla_{\theta^\mu} J = \mathbb{E}\Big[\nabla_a Q_{\theta^Q}(s,a)\big|_{a=\mu_{\theta^\mu}(s)}\, \nabla_{\theta^\mu} \mu_{\theta^\mu}(s)\Big].$$

6. Use **experience replay** (random mini-batches) and **target networks** (slow update) to stabilize training.

Maintaining Stability in DDPG

- **Experience Replay**: store past transitions (s_t, a_t, r_t, s_{t+1}) in a buffer; sample randomly to break correlation.
- **Target Networks**:

$$\theta^{Q'} \leftarrow \tau\,\theta^{Q} + (1-\tau)\,\theta^{Q'},$$
$$\theta^{\mu'} \leftarrow \tau\,\theta^{\mu} + (1-\tau)\,\theta^{\mu'},$$

  ensuring the critic does not overfit to recent data.
- Final output: a *policy* that adjusts portfolio weights dynamically in response to evolving market conditions.

Key Takeaways on DDPG

- Designed for continuous action spaces $\implies$ suitable for portfolio allocation problems.
- **Actor-Critic** method allows simultaneous learning of:
  - A policy function (actor).
  - A value function (critic).
- Stabilization via experience replay and target networks is critical in highly stochastic financial environments.
- Allows the agent to learn trading strategies endogenously, without prior structural knowledge of price formation.

Investigation Strategy

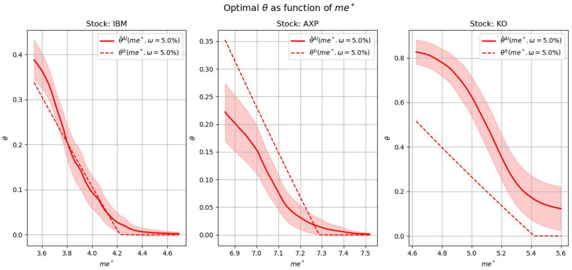
Training Setup:

- 50 simulations of 500 episodes each, each episode has 95 time periods.
- Each AI agent invests in one risky asset and a risk-free bond.
  - Set of stocks for this presentation: $\mathcal{N} = \{\text{IBM}, \text{AXP}, \text{KO}\}$.
- State space is $\mathcal{I}_t = \{me^*_{n,t}, \{x_{n,k,t}\}_{k=1}^{K-1}\}$, action space is $\theta_n \in [0,1]$.
- For each stock $n \in \mathcal{N}$ we consider:
  - $J \in \{1, 2, 5\}$ AI traders.
  - AI traders' aggregate size: $\omega J \in \{1\%, 5\%, 10\%\}$ of stock's mkt cap.

Comparisons:

- Compare average AI weight $\theta^{AI} = \frac{\sum_{j=1}^{J} \theta^j}{J}$ vs. RE benchmark θ^b.
- Compute the distance in policy functions for (i) average AI vs benchmark $\mathrm{E}\left(\left|\theta^{AI} - \theta^b\right|\right)$, and (ii) across AIs $\mathrm{E}\left(\left|\theta^i - \theta^j\right|\right)$.
- Evaluate market efficiency $\mathcal{ME}$ and liquidity $\mathcal{L}$ against RE benchmark.

Portfolio Policies: $J = 1, \omega = 5\%$

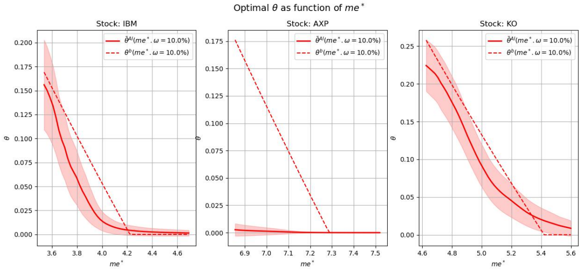


Average θ^{AI} across simulations (solid line) vs. θ^{b} (dashed line) against me^*.

Key Points:

- θ^{AI} decreases in me^* as in Proposition 1.
- A higher me^* given $\{x_n\}$ reflects a larger inferred latent demand, which signals lower expected capital gains (mean reversion).

Portfolio Policies: $J = 1, \omega = 10\%$

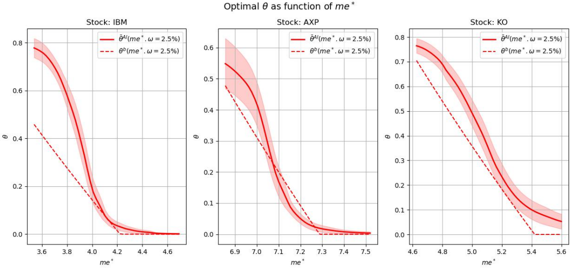


Average θ^{AI} across simulations (solid line) vs. θ^{b} (dashed line) against me^{*}.

Key Points:

- Larger size (ω) leads to more cautious holdings.
- AI internalizes its greater price impact, consistent with Proposition 1.

Portfolio Policies: $J = 2, J\omega = 5\%$

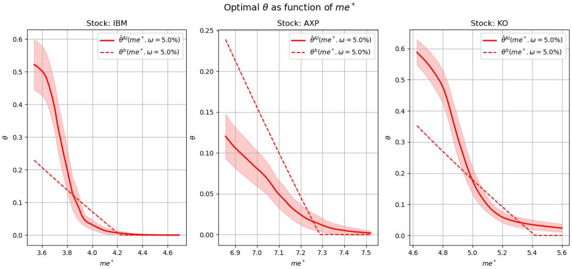


Average θ^{AI} across simulations (solid line) vs. θ^{b} (dashed line) against me^{*}.

Key Points:

- Two AI traders, each manages 2.5% of the market cap. of the asset.
- Policy is more aggressive than benchmark on average, consistent with limited internalization of price impact.

Portfolio Policies: $J = 2, J\omega = 10\%$

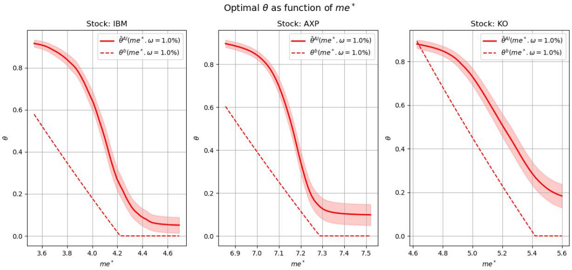


Average θ^{AI} across simulations (solid line) vs. θ^{b} (dashed line) against me^{*}.

Key Points:

- Two AI traders, each manages 5% of the market cap. of the asset.
- The AI reduces its holdings as size (ω) increases, but not as sharply compared to the benchmark.
- Deviation from rational benchmark is more pronounced.

Portfolio Policies: $J = 5, J\omega = 5\%$

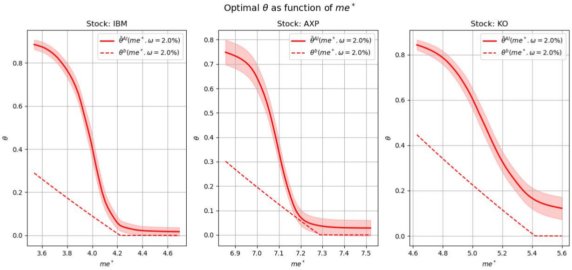


Average θ^{AI} across simulations (solid line) vs. θ^{b} (dashed line) against me^{*}.

Key Points:

- Five AI traders, each with 1% of the market cap. of the asset.
- AI is significantly more aggressive compared to the benchmark.

Portfolio Policies: $J = 5, J\omega = 10\%$



Average θ^{AI} across simulations (solid line) vs. θ^b (dashed line) against me^*.

Key Points:

- Five AI traders, each with 2% of the market cap. of the asset.
- Even larger divergences from the rational line.

Interpretation of Portfolio Policies

Effect of J and ω:

- Size (ω): AI reduces demand but not as sharply as the benchmark.
- Competition (J): AI is more aggressive compared to the benchmark.

Qualitative vs. Quantitative Match:

- Qualitatively, the AI's policy mirrors the benchmark's comparative statics in Proposition 1.
- Quantitatively, the AI falls short when $J > 1$, and the deviation from the benchmark increases in J.

Distances Between AI and Benchmark, $\mathrm{E}\left(\left|\theta^{AI} - \theta^{b}\right|\right)$

	$J\omega = 1\%$			$J\omega = 5\%$			$J\omega = 10\%$		
	IBM	AMX	KO	IBM	AMX	KO	IBM	AMX	KO
J=1	0.149	0.224	0.199	0.068	0.118	0.239	0.046	0.088	0.07
	(0.07)	(0.064)	(0.057)	(0.009)	(0.033)	(0.124)	(0.011)	(0.008)	(0.016)
J=2	0.155	0.256	0.219	0.136	0.14	0.12	0.087	0.086	0.103
	(0.062)	(0.086)	(0.065)	(0.025)	(0.022)	(0.039)	(0.031)	(0.018)	(0.017)
J=5	0.186	0.248	0.219	0.273	0.231	0.196	0.264	0.186	0.235
	(0.060)	(0.030)	(0.038)	(0.058)	(0.064)	(0.062)	(0.042)	(0.067)	(0.07)

- J=1: distances can be quite small, indicating near-benchmark behavior.
- For fixed $J\omega$, the distance with the benchmark increases with J.

Interpretation: a Negative Learning Externality

- Multiple AI traders interfere with each other's learning, leading to larger deviations from the benchmark.

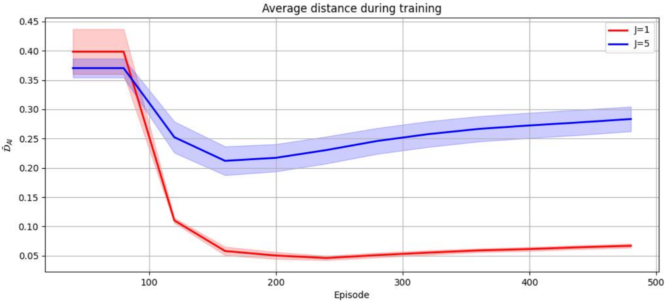


Figure: Average $\left|\theta^{AI} - \theta^{b}\right|$ during training

Average Returns for AIs vs. Benchmark

Panel A: IBM

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.02128 (0.00609)	0.0238	0.01300 (0.00051)	0.0146	0.0102 (0.00101)	0.01201
J=2	0.02036 (0.00434)	0.02285	0.00464 (0.00268)	0.01362	0.0035 (0.00494)	0.01177
J=5	0.01816 (0.00428)	0.02218	-0.01277 (0.00666)	0.01158	-0.03019 (0.00702)	0.01085

Panel B: AXP

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.02762 (0.0088)	0.03304	0.01443 (0.0031)	0.01857	0.00886 (0.00081)	0.0143
J=2	0.02144 (0.01207)	0.03193	0.01268 (0.00269)	0.01652	0.01183 (0.00176)	0.01342
J=5	0.02185 (0.00423)	0.03116	-0.00697 (0.00878)	0.01345	-0.01426 (0.01348)	0.01123

Panel C: KO

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.04195 (0.01168)	0.04623	0.00076 (0.0291)	0.02782	0.01874 (0.00163)	0.02138
J=2	0.03546 (0.01203)	0.04506	0.01848 (0.00585)	0.02254	0.00712 (0.00336)	0.01782
J=5	0.03634 (0.0075)	0.04420	0.00128 (0.01067)	0.01687	-0.02627 (0.01505)	0.01172

Interpretation: Average Returns

- $J = 1$: the single-agent setting achieves near-rational performance.
- As J rises, average returns drop faster than in the benchmark.
- For $J = 5$ and large ω, the AI's mean returns turn negative.
- Overly aggressive strategies and learning interference can sharply degrade performance.

Market Efficiency $\mathcal{M}(\omega, J)$

Panel A: IBM, $\mathcal{M}(\omega, 0) = 0.942$

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.955 (0.002)	0.958	0.965 (0.003)	0.969	0.958 (0.01)	0.969
J=2	0.956 (0.002)	0.968	0.979 (0.002)	0.976	0.972 (0.005)	0.976
J=5	0.955 (0.002)	0.980	0.984 (0.003)	0.980	0.973 (0.005)	0.981

Panel B: AXP, $\mathcal{M}(\omega, 0) = 0.889$

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.905 (0.003)	0.912	0.912 (0.013)	0.940	0.890 (0.006)	0.941
J=2	0.903 (0.005)	0.929	0.931 (0.013)	0.951	0.914 (0.013)	0.954
J=5	0.903 (0.002)	0.957	0.943 (0.006)	0.957	0.946 (0.004)	0.963

Panel C: KO, $\mathcal{M}(\omega, 0) = 0.906$

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.917 (0.002)	0.920	0.946 (0.010)	0.952	0.940 (0.006)	0.955
J=2	0.915 (0.003)	0.933	0.950 (0.004)	0.957	0.965 (0.002)	0.966
J=5	0.915 (0.002)	0.951	0.947 (0.007)	0.959	0.970 (0.007)	0.973

Interpretation: Market Efficiency

- AI traders increase efficiency vs. no-AI baseline.
- Competition (higher J for fixed $J\omega$) increases market efficiency via more aggressive trading, but not as much as in the RE benchmark.
- But $\mathcal{ME}$ exceeds the RE benchmark only occasionally. Why?
- With RE, more aggressive trading always pushes $Var\left[E_t(R^e_{n,t+1})\right]$ down, but not with AI traders because of imperfect learning.

Market Liquidity $\mathcal{L}^{AI}(\omega, J)$

Panel B: IBM

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.006 (0.001)	0.052	0.013 (0.002)	0.096	0.009 (0.006)	0.096
J=2	0.006 (0.001)	0.054	0.02 (0.004)	0.128	0.006 (0.005)	0.129
J=5	0.006 (0.001)	0.054	0.011 (0.008)	0.154	-0.002 (0.005)	0.16

Panel B: AXP

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.010 (0.002)	0.037	0.015 (0.008)	0.091	0.001 (0.004)	0.092
J=2	0.009 (0.004)	0.038	0.028 (0.01)	0.116	0.014 (0.007)	0.123
J=5	0.009 (0.001)	0.039	0.015 (0.015)	0.133	0.003 (0.015)	0.153

Panel C: KO

	$J\omega$=1% AI	$J\omega$=1% RB	$J\omega$=5% AI	$J\omega$=5% RB	$J\omega$=10% AI	$J\omega$=10% RB
J=1	0.004 (0.001)	0.026	0.019 (0.005)	0.095	0.015 (0.004)	0.105
J=2	0.003 (0.001)	0.026	0.013 (0.003)	0.110	0.001 (0.003)	0.139
J=5	0.006 (0.001)	0.026	0.003 (0.008)	0.116	-0.006 (0.005)	0.165

Interpretation: Liquidity

- Liquidity is lower than the RE benchmark by an order of magnitude.
- Under RE, speculators recognize a temporary supply shock and buy aggressively, anticipating mean reversion.
- AI traders misinterpret price drops as weakened latent demand, failing to fully exploit short-term arbitrage.
- As a result, they provide liquidity well below rational levels.
- More generally, this suggests that AI trading may not react optimally to out-of-equilibrium shocks or regime changes.

Conclusion

- AI traders learn return predictability and impact efficiency and liquidity.
- Qualitative alignment with theory:
  - Portfolio weights decline in me^* (Prop. 1).
  - More AI capital reduces predictability (Prop. 2).
  - Liquidity improves but stays below the rational benchmark (Prop. 3).
- Learning externality:
  - Multiple AIs distort learning, limiting price impact internalization.
  - This lowers AI profits and weakens market efficiency and liquidity.
- Future work: dynamic rebalancing (additional state variables), multiple assets (cross sectional predictability).

Abada, I. and X. Lambin (2023): "Artificial Intelligence: Can Seemingly Collusive Outcomes Be Avoided?" *Management Science*, 69, 5042–5065.

Bai, J., T. Philippon, and A. Savov (2016): "Have Financial Markets Become More Informative?" *Journal of Financial Economics*, 122, 625–654.

Barberis, N. and L. Jin (2023): "Model-free and Model-based Learning as Joint Drivers of Investor Behavior," Available at `https://www.law.nyu.edu/sites/default/files/DGJ_AI_Trading_Collusion_Market%20Manipulation_Penn_NYU_Law.pdf`.

Biais, B., T. Foucault, and S. Moinas (2015): "Equilibrium Fast Trading," *Journal of Financial Economics*, 116, 292–313.

Brogaard, J., T. Hendershott, and R. Riordan (2015): "Quantifying the High-Frequency Trading "Arms Race"," *Quarterly Journal of Economics*, 130, 1547–1621.

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2020): "Artificial Intelligence, Algorithmic Pricing, and Collusion," *American Economic Review*, 110, 3267–3297.

Colliard, J.-E., T. Focault, and S. Lovo (2023): "Algorithmic Pricing and Liquidity in Securities Markets," Working Paper.

Dou, W. W., I. Goldstein, and Y. Ji (2023): "AI-Powered Trading, Algorithmic Collusion, and Price Efficiency," Available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4452704.

Dugast, J. and T. Foucault (2018): "Data abundance and asset price informativeness," *Journal of Financial Economics*, 130, 367–391.

Farbood, M., A. Matray, L. Veldkamp, and V. Venkateswaran (forthcoming): "Where Has All the Data Gone?" *Review of Financial Studies*.

Foucault, T., J. Hombert, and I. Roşu (2016): "News Trading and Speed," *Journal of Finance*, 71, 335–382.

Johnson, J. P., A. Rhodes, and M. Wildenbeest (2023): "Platform Design When Sellers Use Pricing Algorithms," *Econometrica*, 91, 1841–1879.

Koijen, R. S. J. and M. Yogo (2019): "A Demand System Approach to Asset Pricing," Forthcoming at *Journal of Political Economy*.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015): "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*.

Vyetrenko, S. and S. Xu (2019): "Risk-Sensitive Compact Decision Trees for Autonomous Execution in Presence of Simulated Market Response," *ArXiv*, abs/1906.02312.

WATKINS, C. J. C. H. (1989): "Learning from Delayed Rewards," Ph.D. thesis, King's College, Cambridge.

YANG, H. (2024): "AI Coordination and Self-Fulfilling Financial Crises," Available at https://drive.google.com/file/d/1SCb68JxyXN3Zq35dXrU3ZHs8Ruoxrce-/view.

YANG, H., X.-Y. LIU, S. ZHONG, AND A. WALID (2020): "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," Tech. rep.

ABADA, I. AND X. LAMBIN (2023): "Artificial Intelligence: Can Seemingly Collusive Outcomes Be Avoided?" *Management Science*, 69, 5042–5065.

BAI, J., T. PHILIPPON, AND A. SAVOV (2016): "Have Financial Markets Become More Informative?" *Journal of Financial Economics*, 122, 625–654.

BARBERIS, N. AND L. JIN (2023): "Model-free and Model-based Learning as Joint Drivers of Investor Behavior," Available at https://www.law.nyu.edu/sites/default/files/DGJ_AI_Trading_Collusion_Market%20Manipulation_Penn_NYU_Law.pdf.

BIAIS, B., T. FOUCAULT, AND S. MOINAS (2015): "Equilibrium Fast Trading," *Journal of Financial Economics*, 116, 292–313.

BROGAARD, J., T. HENDERSHOTT, AND R. RIORDAN (2015): "Quantifying the High-Frequency Trading "Arms Race"," *Quarterly Journal of Economics*, 130, 1547–1621.

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2020): "Artificial Intelligence, Algorithmic Pricing, and Collusion," *American Economic Review*, 110, 3267–3297.

Colliard, J.-E., T. Focault, and S. Lovo (2023): "Algorithmic Pricing and Liquidity in Securities Markets," Working Paper.

Dou, W. W., I. Goldstein, and Y. Ji (2023): "AI-Powered Trading, Algorithmic Collusion, and Price Efficiency," Available at `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4452704`.

Dugast, J. and T. Foucault (2018): "Data abundance and asset price informativeness," *Journal of Financial Economics*, 130, 367–391.

Farbood, M., A. Matray, L. Veldkamp, and V. Venkateswaran (forthcoming): "Where Has All the Data Gone?" *Review of Financial Studies*.

Foucault, T., J. Hombert, and I. Roşu (2016): "News Trading and Speed," *Journal of Finance*, 71, 335–382.

Johnson, J. P., A. Rhodes, and M. Wildenbeest (2023): "Platform Design When Sellers Use Pricing Algorithms," *Econometrica*, 91, 1841–1879.

Koijen, R. S. J. and M. Yogo (2019): "A Demand System Approach to Asset Pricing," Forthcoming at *Journal of Political Economy*.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015): "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*.

Vyetrenko, S. and S. Xu (2019): "Risk-Sensitive Compact Decision Trees for Autonomous Execution in Presence of Simulated Market Response," *ArXiv*, abs/1906.02312.

Watkins, C. J. C. H. (1989): "Learning from Delayed Rewards," Ph.D. thesis, King's College, Cambridge.

Yang, H. (2024): "AI Coordination and Self-Fulfilling Financial Crises," Available at `https://drive.google.com/file/d/1SCb68JxyXN3Zq35dXrU3ZHs8Ruoxrce-/view`.

Yang, H., X.-Y. Liu, S. Zhong, and A. Walid (2020): "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," Tech. rep.