AI-Powered Trading, Algorithmic Collusion, and Price Efficiency

Winston W. Dou[◇] Itay Goldstein[◇] Yan Ji[†]

[◇]University of Pennsylvania and NBER

[†]Hong Kong University of Science and Technology

IESE Banking Initiative Workshop

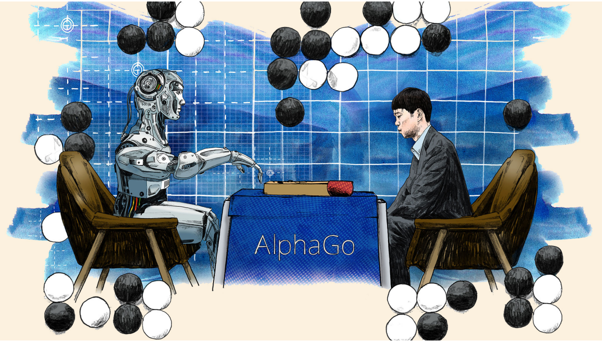
Spring, 2025

What is “AI-powered trading?”

AI-powered trading:

Algorithmic trading system **+** **reinforcement-learning (“RL”) algorithms**

RL algo is a key approach of AI, and serves as the backbone of “AlphaGo”



Note: # possible legal moves ($\approx 10^{170}$) $\gg$ # atoms in the universe ($\approx 10^{80}$)

RL-backed AI (pattern recognition) $\gg$ human cognitive (logical thinking) for complex tasks

RL algorithms are model-free and self-learning

A multi-agent system, where each agent is indexed by i and solves

$$V_i(s) = \max_{x_i \in \mathcal{X}} \{\mathbb{E}[u_i | s, x_i] + \rho \mathbb{E}[V_i(s') | s, x_i]\}, \quad \text{where } i = 1, \cdots, I,$$

- s = state in current period, and s' = state in next period
- ρ = discount factor
- u_i = payoff of agent i, also depending on the actions of other agents x_{-i}

RL algorithms solve the Bellman equation on a model-free, self-learning basis, without assuming

- The system is already in equilibrium
- Agents know the true distribution of states and payoffs

RL algorithms are model-free and self-learning

A multi-agent system, where each agent is indexed by i and solves

$$V_i(s) = \max_{x_i \in \mathcal{X}} \left\{ \mathbb{E}\left[u_i | s, x_i\right] + \rho \mathbb{E}\left[V_i(s') | s, x_i\right] \right\}, \quad \text{where } i = 1, \cdots, I,$$

- $s =$ state in current period, and $s' =$ state in next period
- $\rho =$ discount factor
- $u_i =$ payoff of agent i, also depending on the actions of other agents x_{-i}

RL algorithms solve the Bellman equation on a model-free, self-learning basis, without assuming

- The system is already in equilibrium
- Agents know the true distribution of states and payoffs

Q-learning: A foundation of numerous RL algorithms

$\boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x_i}) =$ value function of agent i when taking action x_i in state s

Note: Dynamically sophisticated by tracing endogenous state transitions, unlike bandit algorithms

$\boldsymbol{V_i}(\boldsymbol{s}) = \max_{\boldsymbol{x}' \in \mathcal{X}} \boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x}')$, **with Q_i's recursive relation:**

$$Q_i(s, x_i) \equiv \mathbb{E}\left[u_i + \rho \max_{x' \in \mathcal{X}} Q_i(s', x') \middle| s, x_i\right]$$

Estimate $Q_i(s, x)$ through $\widehat{Q}_{i,t}(s, x)$, employing $\widehat{Q}_{i,t}$'s recursive updating:

$$\widehat{Q}_{i,t+1}(s_t, x_{i,t}) = \underbrace{\alpha\left[u_{i,t} + \rho \max_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_{t+1}, x')\right]}_{\text{new experimental data}} + \underbrace{(1-\alpha)\widehat{Q}_{i,t}(s_t, x_{i,t})}_{\text{previous learning}}$$

The update of $\widehat{Q}_{i,t+1}$ takes place at $(\boldsymbol{s_t}, \boldsymbol{x_{i,t}})$, where $x_{i,t}$ is chosen as:

$$x_{i,t} = \begin{cases} \operatorname{argmax}_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_t, x'), & \text{with prob. } 1 - \varepsilon_t \quad \text{(exploitation)} \\ \widetilde{x} \sim \text{uniform on } \mathcal{X}, & \text{with prob. } \varepsilon_t \quad \text{(exploration)} \end{cases}$$

Q-learning: A foundation of numerous RL algorithms

$\boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x_i})$ = value function of agent i when taking action x_i in state s

Note: Dynamically sophisticated by tracing endogenous state transitions, unlike bandit algorithms

$\boldsymbol{V_i}(\boldsymbol{s}) = \max_{\boldsymbol{x'} \in \mathcal{X}} \boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x'})$, with $\boldsymbol{Q_i}$'s recursive relation:

$$Q_i(s, x_i) \equiv \mathbb{E}\left[u_i + \rho \max_{x' \in \mathcal{X}} Q_i(s', x') \middle| s, x_i \right]$$

Estimate $\boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x})$ through $\widehat{\boldsymbol{Q}}_{\boldsymbol{i,t}}(\boldsymbol{s}, \boldsymbol{x})$, employing $\widehat{\boldsymbol{Q}}_{\boldsymbol{i,t}}$'s recursive updating:

$$\widehat{Q}_{i,t+1}(s_t, x_{i,t}) = \underbrace{\alpha \left[u_{i,t} + \rho \max_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_{t+1}, x') \right]}_{\text{new experimental data}} + \underbrace{(1-\alpha)\widehat{Q}_{i,t}(s_t, x_{i,t})}_{\text{previous learning}}$$

The update of $\widehat{\boldsymbol{Q}}_{\boldsymbol{i,t+1}}$ takes place at $(\boldsymbol{s_t}, \boldsymbol{x_{i,t}})$, where $x_{i,t}$ is chosen as:

$$x_{i,t} = \begin{cases} \operatorname{argmax}_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_t, x'), & \text{with prob. } 1 - \varepsilon_t \quad \text{(exploitation)} \\ \widetilde{x} \sim \text{uniform on } \mathcal{X}, & \text{with prob. } \varepsilon_t \quad \text{(exploration)} \end{cases}$$

Wharton
University of Pennsylvania

Q-learning: A foundation of numerous RL algorithms

$\boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x_i})$ = value function of agent i when taking action x_i in state s

Note: Dynamically sophisticated by tracing endogenous state transitions, unlike bandit algorithms

$V_i(s) = \max_{x' \in \mathcal{X}} Q_i(s, x')$, with Q_i's recursive relation:

$$Q_i(s, x_i) \equiv \mathbb{E}\left[u_i + \rho \max_{x' \in \mathcal{X}} Q_i(s', x') \middle| s, x_i\right]$$

Estimate $Q_i(s, x)$ through $\widehat{Q}_{i,t}(s, x)$, employing $\widehat{Q}_{i,t}$'s recursive updating:

$$\widehat{Q}_{i,t+1}(s_t, x_{i,t}) = \underbrace{\alpha\left[u_{i,t} + \rho \max_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_{t+1}, x')\right]}_{\text{new experimental data}} + \underbrace{(1-\alpha)\widehat{Q}_{i,t}(s_t, x_{i,t})}_{\text{previous learning}}$$

The update of $\widehat{Q}_{i,t+1}$ takes place at $(s_t, x_{i,t})$, where $x_{i,t}$ is chosen as:

$$x_{i,t} = \begin{cases} \operatorname{argmax}_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_t, x'), & \text{with prob. } 1-\varepsilon_t \quad \text{(exploitation)} \\ \widetilde{x} \sim \text{uniform on } \mathcal{X}, & \text{with prob. } \varepsilon_t \quad \text{(exploration)} \end{cases}$$

Wharton UNIVERSITY of PENNSYLVANIA

Q-learning: A foundation of numerous RL algorithms

$\boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x_i})$ = value function of agent i when taking action x_i in state s

Note: Dynamically sophisticated by tracing endogenous state transitions, unlike bandit algorithms

$\boldsymbol{V_i}(\boldsymbol{s}) = \max_{\boldsymbol{x}' \in \mathcal{X}} \boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x}')$, with $\boldsymbol{Q_i}$'s recursive relation:

$$Q_i(s, x_i) \equiv \mathbb{E}\left[u_i + \rho \max_{x' \in \mathcal{X}} Q_i(s', x') \middle| s, x_i\right]$$

Estimate $\boldsymbol{Q_i}(\boldsymbol{s}, \boldsymbol{x})$ through $\widehat{\boldsymbol{Q}}_{\boldsymbol{i},\boldsymbol{t}}(\boldsymbol{s}, \boldsymbol{x})$, employing $\widehat{\boldsymbol{Q}}_{\boldsymbol{i},\boldsymbol{t}}$'s recursive updating:

$$\widehat{Q}_{i,t+1}(s_t, x_{i,t}) = \underbrace{\alpha \left[u_{i,t} + \rho \max_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_{t+1}, x')\right]}_{\text{new experimental data}} + \underbrace{(1-\alpha)\widehat{Q}_{i,t}(s_t, x_{i,t})}_{\text{previous learning}}$$

The update of $\widehat{\boldsymbol{Q}}_{\boldsymbol{i},\boldsymbol{t}+\boldsymbol{1}}$ takes place at $(\boldsymbol{s_t}, \boldsymbol{x_{i,t}})$, where $x_{i,t}$ is chosen as:

$$x_{i,t} = \begin{cases} \operatorname{argmax}_{x' \in \mathcal{X}} \widehat{Q}_{i,t}(s_t, x'), & \text{with prob. } 1 - \varepsilon_t \quad \text{(exploitation)} \\ \widetilde{x} \sim \text{uniform on } \mathcal{X}, & \text{with prob. } \varepsilon_t \quad \text{(exploration)} \end{cases}$$

Outline

1. Motivation

2. Laboratory framework & theoretical benchmark

3. Simulation experiments

- Q-learning algorithms in trading
- Experimental configuration and setup
- Simulation results

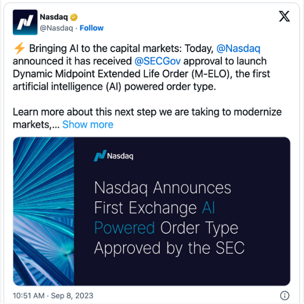
Rise of AI in financial and retail markets

SEC approves Nasdaq's AI trading system

- Using RL algos to facilitate AI trading

Other examples:

- FX digital trading platforms (e.g., MetaTrader)
- Crypto trading platforms



AI pricing algos in e-commerce, gasoline, and housing rental markets

e.g., Chen_Mislove_Wilson (2016), Brown_Mackay (2023), Assad_Clark_Ershov_Xu (2023)

- Notably, "AI collusion" has emerged as a new potential antitrust challenge
- Definition: Autonomous self-interested algos learn to achieve and maintain coordination without agreement, communication, or even intention
- Lawsuits were filed, and congress was urged to reform Antitrust Law

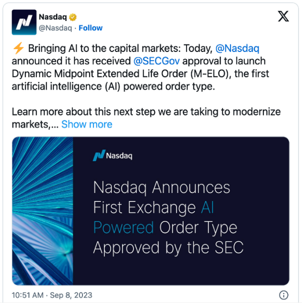
Rise of AI in financial and retail markets

SEC approves Nasdaq's AI trading system

- Using RL algos to facilitate AI trading

Other examples:

- FX digital trading platforms (e.g., MetaTrader)
- Crypto trading platforms



AI pricing algos in e-commerce, gasoline, and housing rental markets

e.g., Chen_Mislove_Wilson (2016), Brown_Mackay (2023), Assad_Clark_Ershov_Xu (2023)

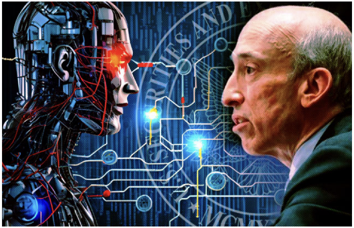
- Notably, "AI collusion" has emerged as a new potential antitrust challenge
- Definition: Autonomous self-interested algos learn to achieve and maintain coordination without agreement, communication, or even intention
- Lawsuits were filed, and congress was urged to reform Antitrust Law

SEC: Risk of AI-driven market manipulation?

SEC Chair, Gary Gensler, has warned that

"Financial market instability, or even a financial crisis, caused by AI is nearly unavoidable without regulation.

"Even if the humans aren't talking, the machines will start to have a sense of cooperation. We've already seen this in high-frequency trading."



This paper: "AI collusion" can robustly arise through two distinct mechanisms, undermining competition and market efficiency

- **Market liquidity** ↓
  - ⇒ Funding liquidity ↓ ⇒ financial market instability ↑ (real effects, existing studies)
- **Price informativeness** ↓ **+ mispricing** ↑
  - ⇒ Distortion in real decisions ↑ ⇒ fundamental value ↓ (real effects, existing studies)

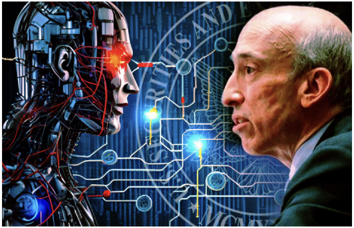
Our approach: A proof-of-concept experimental study on AI trading algos

SEC: Risk of AI-driven market manipulation?

SEC Chair, Gary Gensler, has warned that

"Financial market instability, or even a financial crisis, caused by AI is nearly unavoidable without regulation.

"Even if the humans aren't talking, the machines will start to have a sense of cooperation. We've already seen this in high-frequency trading."



This paper: "AI collusion" can robustly arise through two distinct mechanisms, undermining competition and market efficiency

- **Market liquidity** ↓
  - ⇒ Funding liquidity ↓ ⇒ financial market instability ↑ (real effects, existing studies)
- **Price informativeness ↓ + mispricing ↑**
  - ⇒ Distortion in real decisions ↑ ⇒ fundamental value ↓ (real effects, existing studies)

Our approach: A proof-of-concept experimental study on AI trading algos

Outline

1. **Motivation**

2. **Laboratory framework & theoretical benchmark**

3. **Simulation experiments**
   - Q-learning algorithms in trading
   - Experimental configuration and setup
   - Simulation results

Extend "static" Kyle (1985) to a repeated-trading setting

Within each period t:

(1) Fundamental value of an asset: $v_t \sim^{i.i.d.} N(\overline{v}, \sigma_v^2)$

A continuum of noise traders place a collective order flow: $u_t \sim^{i.i.d.} N(0, \sigma_u^2)$

(2) Each of I oligopolistic informed speculator i knows v_t (not u_t) and solves

$$V_i(s_t) = \max_{x_{i,t}} \mathbb{E}\left[(v_t - p_t)x_{i,t} + \rho V_i(s_{t+1}) | s_t, x_{i,t}\right],$$

where p_t is market price, and s_t includes v_t and public information before t

(3) A continuum of information-insensitive investors with a demand curve:

$$z_t = -\xi(p_t - \overline{v}), \quad \text{with } \xi > 0, \qquad \text{(e.g., Kyle\_Xiong, 2001)}$$

(4) A market maker observes $y_t = \sum_{i=1}^{I} x_{i,t} + u_t$ and knows the z_t schedule, then determines p_t as follows:

$$\min_{p_t} \underbrace{(y_t + z_t)^2}_{\text{"inventory costs"}} + \theta \underbrace{\mathbb{E}[(p_t - v_t)^2 | y_t]}_{\text{"pricing errors"}}, \quad \text{with } \theta > 0 \text{ and } \theta \approx 0$$

Wharton University of Pennsylvania

Extend "static" Kyle (1985) to a repeated-trading setting

Within each period t:

(1) Fundamental value of an asset: $v_t \sim^{i.i.d.} N(\overline{v}, \sigma_v^2)$

A continuum of noise traders place a collective order flow: $u_t \sim^{i.i.d.} N(0, \sigma_u^2)$

(2) Each of I oligopolistic informed speculator i knows v_t (not u_t) and solves

$$V_i(s_t) = \max_{x_{i,t}} \mathbb{E}\left[(v_t - p_t)x_{i,t} + \rho V_i(s_{t+1}) | s_t, x_{i,t}\right],$$

where p_t is market price, and s_t includes v_t and public information before t

(3) A continuum of information-insensitive investors with a demand curve:

$$z_t = -\xi(p_t - \overline{v}), \text{ with } \xi > 0, \qquad \text{(e.g., Kyle\_Xiong, 2001)}$$

(4) A market maker observes $y_t = \sum_{i=1}^{I} x_{i,t} + u_t$ and knows the z_t schedule, then determines p_t as follows:

$$\min_{p_t} \underbrace{(y_t + z_t)^2}_{\text{"inventory costs"}} + \theta \underbrace{\mathbb{E}[(p_t - v_t)^2 | y_t]}_{\text{"pricing errors"}}, \quad \text{with } \theta > 0 \text{ and } \theta \approx 0$$

Theoretical benchmarks

Non-collusive Nash equilibrium (*N*)

Speculators do not internalize the impact of their trading on others' profits

Perfect cartel benchmark (*M*)

Speculators collaborate to trade as a unified monopoly, then split the order flow

Collusive equilibrium (*C*)

Speculators reach and sustain a steady state characterized by two properties:

- Agents achieve collective supra-competitive profits
- Agents have the option to deviate from equilibrium actions for short-term gains, and such deviations impose costs on others

Two mechanisms for collusive equilibrium in theory

1. Collusive (Nash) equilibrium through price-trigger strategies

(akin to Green_Porter, 1984)

Speculators adopt "conservative" trading strategy $x_{i,t}^C = \chi^C(v_t - \overline{v})$, anticipating

$$\text{Expected } p_t^C = \overline{v} + \varphi^C(v_t - \overline{v})$$

Once p_t deviates significantly from the expected p_t^C, speculators revert to the non-collusive Nash equilibrium for T periods with probability η each period

2. Collusive (experience-based) equilibrium through learning bias

(akin to Fudenberg_Levine, 1993; Fershtman_Pakes, 2012)

Speculators adopt "conservative" trading strategy $x_{i,t}^C = \chi^C(v_t - \overline{v})$, believing

$$\chi^C = \text{optimal trading strategy due to biased evaluations}$$

Self-confirming belief is correct on the equilibrium path but incorrect off it

Two mechanisms for collusive equilibrium in theory

1. Collusive (Nash) equilibrium through price-trigger strategies

(akin to Green_Porter, 1984)

Speculators adopt "conservative" trading strategy $x_{i,t}^{C} = \chi^{C}(v_t - \overline{v})$, anticipating

$$\text{Expected } \; p_t^{C} = \overline{v} + \varphi^{C}(v_t - \overline{v})$$

Once p_t deviates significantly from the expected p_t^{C}, speculators revert to the non-collusive Nash equilibrium for T periods with probability η each period

2. Collusive (experience-based) equilibrium through learning bias

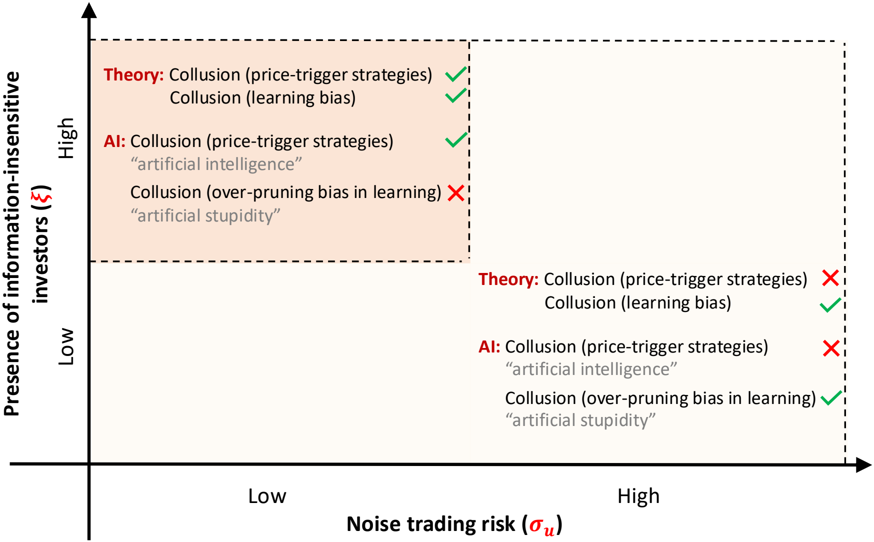
(akin to Fudenberg_Levine, 1993; Fershtman_Pakes, 2012)

Speculators adopt "conservative" trading strategy $x_{i,t}^{C} = \chi^{C}(v_t - \overline{v})$, believing

$$\chi^{C} = \text{optimal trading strategy due to biased evaluations}$$

Self-confirming belief is correct on the equilibrium path but incorrect off it

Theory: Existence of collusive equilibrium



Theoretical properties of price-trigger collusion

Proposition 3 (Collusion through price-trigger strategies):

When I is sufficiently large:

$$\rho \downarrow, \quad \sigma_u \uparrow, \quad \text{or} \quad I \uparrow \quad \Longrightarrow \quad \Delta^C \downarrow, \ \mathcal{I}^C/\mathcal{I}^M \uparrow, \ \mathcal{L}^C/\mathcal{L}^M \uparrow, \ \text{and} \ \ \mathcal{E}^C/\mathcal{E}^M \downarrow$$

Note:

- Δ = trading profitability
- $\mathcal{I}$ = price informativeness
- $\mathcal{L}$ = market liquidity
- $\mathcal{E}$ = mispricing
- C and M represent the collusive equilibrium and cartel benchmark, respectively

Theoretical properties of learning-bias collusion

Proposition 4 (Learning bias due to over-perceived aversion to σ_u):

When ξ is sufficiently large:

$$I \uparrow, \quad \text{or} \quad \sigma_u \downarrow \quad \Longrightarrow \quad \Delta^C \downarrow, \quad \mathcal{I}^C/\mathcal{I}^M \uparrow, \quad \mathcal{L}^C/\mathcal{L}^M \uparrow, \quad \text{and} \quad \mathcal{E}^C/\mathcal{E}^M \downarrow,$$

Furthermore,

$$\rho \quad \text{does not affect} \quad \Delta^C, \quad \mathcal{I}^C/\mathcal{I}^M, \quad \mathcal{L}^C/\mathcal{L}^M, \quad \text{or} \quad \mathcal{E}^C/\mathcal{E}^M$$

Note:

- Δ = trading profitability
- $\mathcal{I}$ = price informativeness
- $\mathcal{L}$ = market liquidity
- $\mathcal{E}$ = mispricing
- *C* and *M* represent the collusive equilibrium and cartel benchmark, respectively

Outline

1. Motivation

2. Laboratory framework & theoretical benchmark

3. Simulation experiments

- Q-learning algorithms in trading
- Experimental configuration and setup
- Simulation results

RL algorithms as experimental subjects

Replace each theoretical speculator i with private information by a Q-learning algorithm $\widehat{Q}_{i,t}(s_t, x_{i,t})$:

- Payoff: $\pi_{i,t} = (v_t - p_t)x_{i,t}$
- State variable: $s_t = \{p_{t-1}, v_{t-1}, v_t\}$
- Exploration rate: $\varepsilon_t = e^{-\beta t}$

Replace the RE market maker with a statistically adaptive market maker

- Linear regressions using "historical data" $\mathcal{D}_t \equiv \{v_{t-\tau}, p_{t-\tau}, z_{t-\tau}, y_{t-\tau}\}_{\tau=1}^{T_m}$
- Results will not change with a Q-learning market maker

RL algorithms as experimental subjects

Replace each theoretical speculator *i* with private information by a Q-learning algorithm $\widehat{Q}_{i,t}(s_t, x_{i,t})$:

- Payoff: $\pi_{i,t} = (v_t - p_t)x_{i,t}$
- State variable: $s_t = \{p_{t-1}, v_{t-1}, v_t\}$
- Exploration rate: $\varepsilon_t = e^{-\beta t}$

Replace the RE market maker with a statistically adaptive market maker

- Linear regressions using "historical data" $\mathcal{D}_t \equiv \{v_{t-\tau}, p_{t-\tau}, z_{t-\tau}, y_{t-\tau}\}_{\tau=1}^{T_m}$
- Results will not change with a Q-learning market maker

Baseline parameter values

Environment parameters:

$$I = 2,\ \sigma_u/\sigma_v = 10^{-1},\ \text{and}\ \ \xi = 500$$

Preference parameters:

$$\rho = 0.95,\ \text{and}\ \ \theta = 0.1$$

Discretization parameters:

$$n_x = 15,\ n_p = 31,\ n_v = 10,\ \text{and}\ \ T_m = 10,000$$

Hyperparameters:

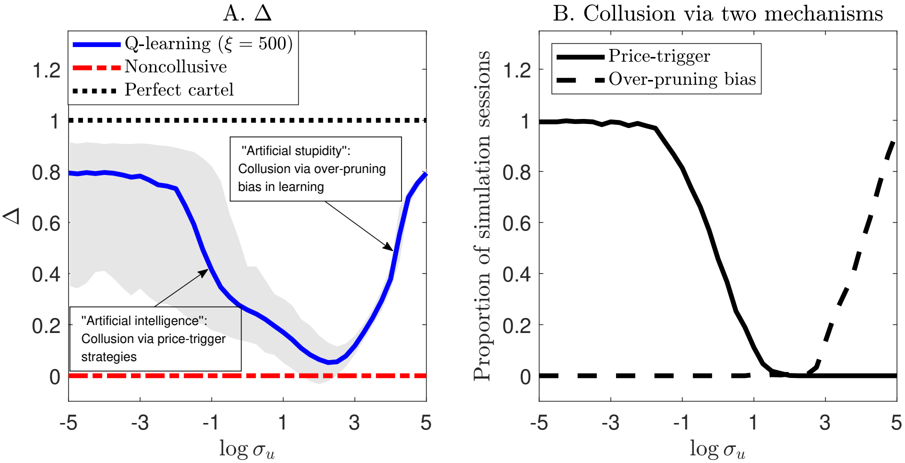
$$\alpha = 0.01\ \ \text{and}\ \ \beta = 5 \times 10^{-7}$$

Note: AI traders do not have prior knowledge of environment parameters

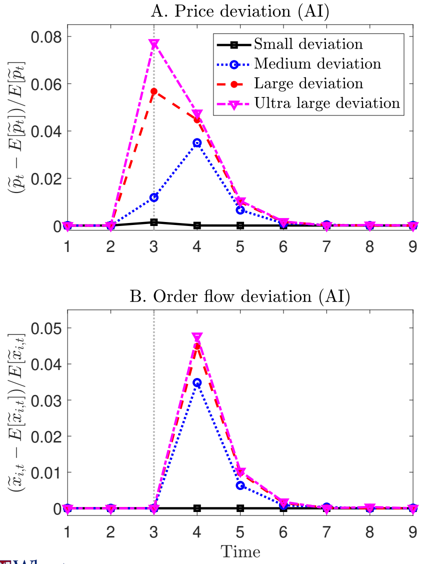
Wharton UNIVERSITY of PENNSYLVANIA

AI Collusion: Two distinct mechanisms

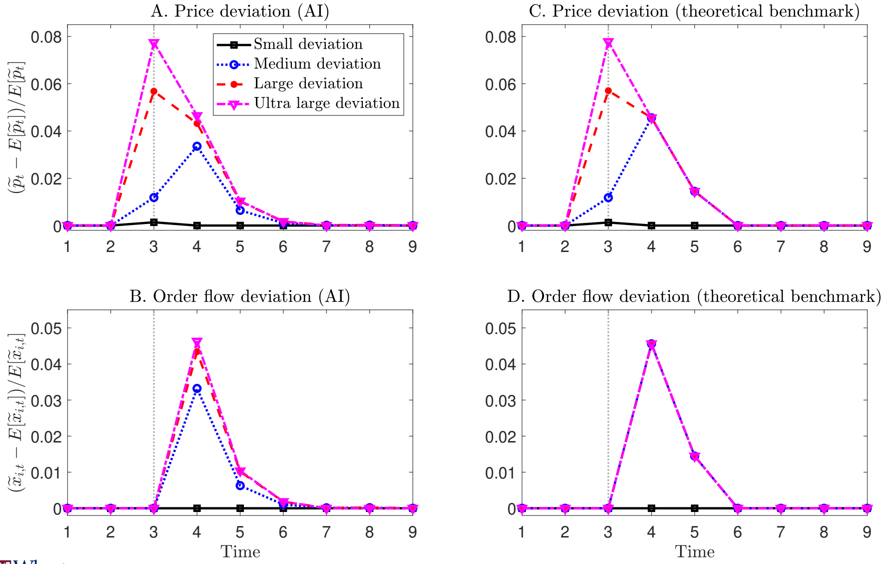
$\Delta = \frac{\pi - \pi^N}{\pi^M - \pi^N}$ captures the collusion profitability, with π = average trading profit



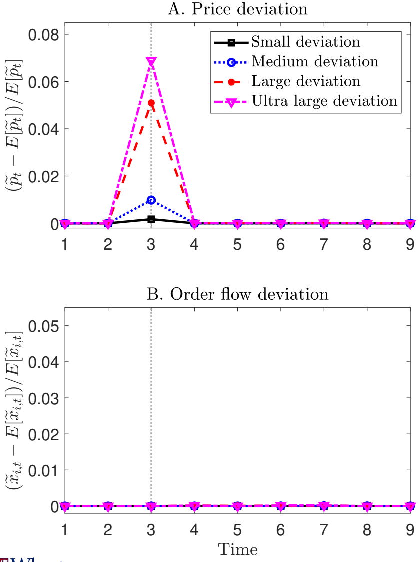
$\left[\xi = 500;\ \sigma_u/\sigma_v = 10^{-1}\right]$: Price-trigger strategies



$\left[\xi = 500;\ \sigma_u/\sigma_v = 10^{-1}\right]$: Price-trigger strategies

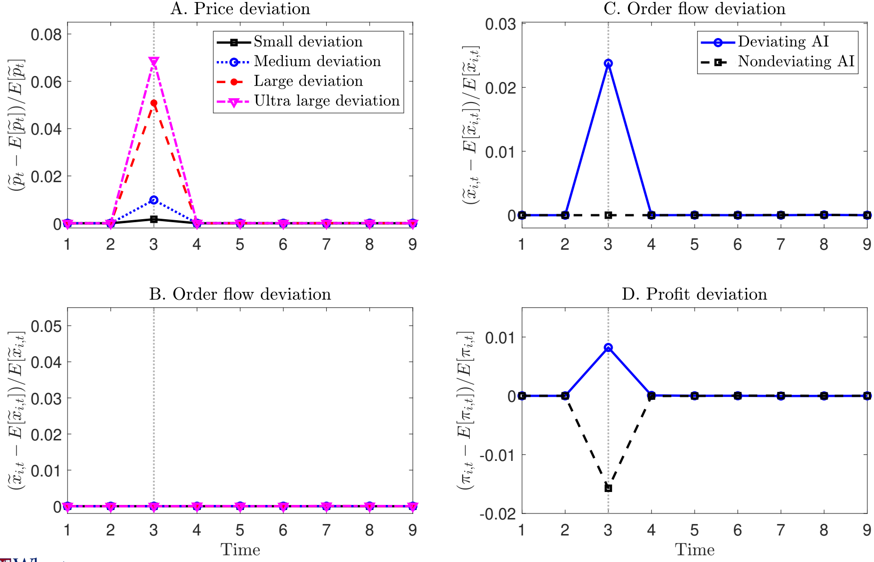


$[\xi = 500; \sigma_u/\sigma_v = 10^2]$ Over-pruning bias in learning



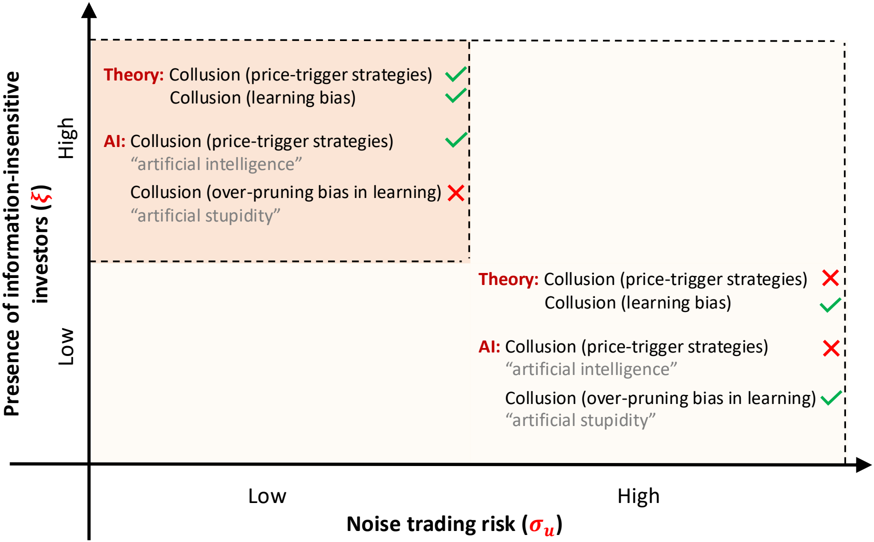
Wharton
UNIVERSITY of PENNSYLVANIA

$[\xi = 500; \sigma_u/\sigma_v = 10^2]$ Over-pruning bias in learning



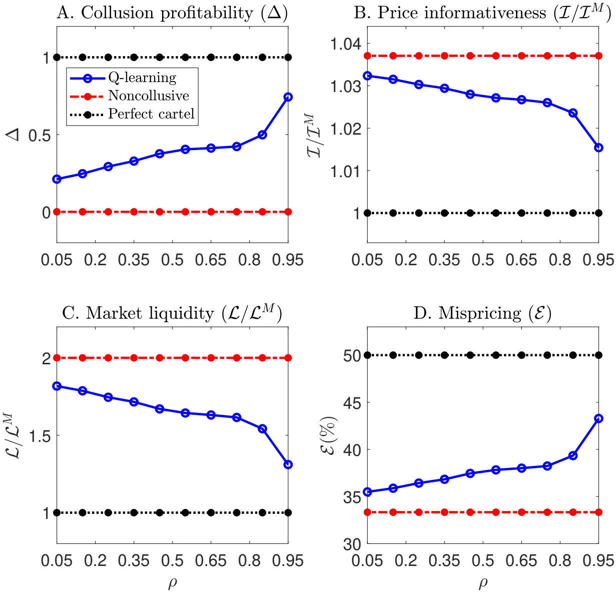
Wharton
University of Pennsylvania

Summary of our main findings

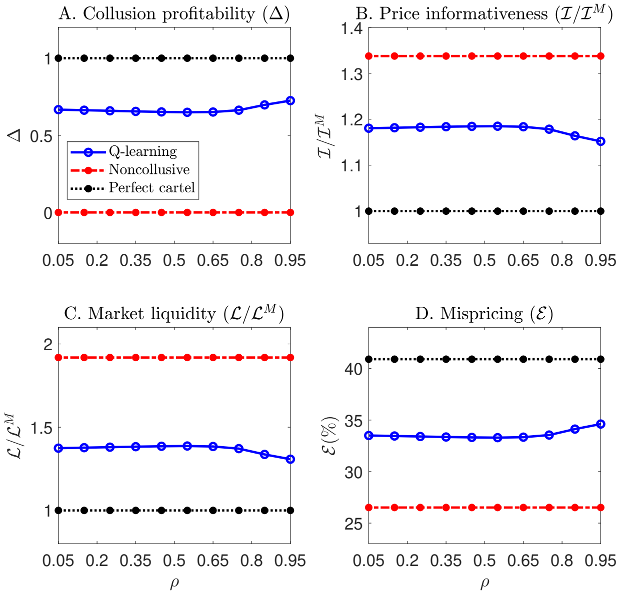


Wharton UNIVERSITY of PENNSYLVANIA

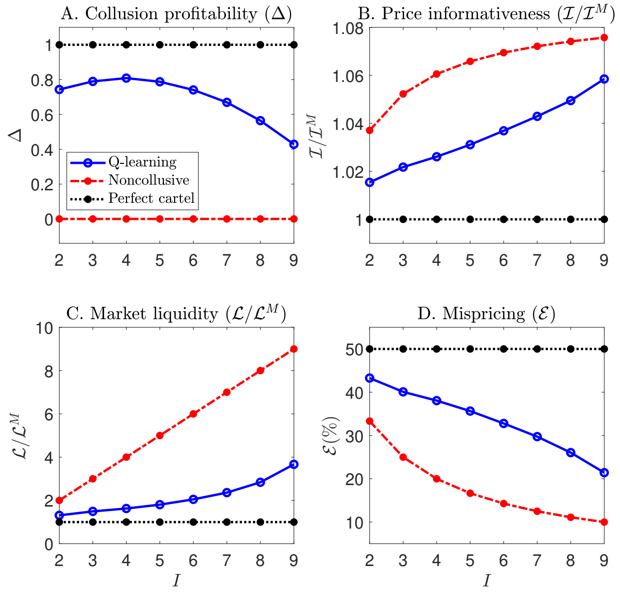
Folk Theorem: Price-trigger strategies ($\sigma_u/\sigma_v = 10^{-1}$)



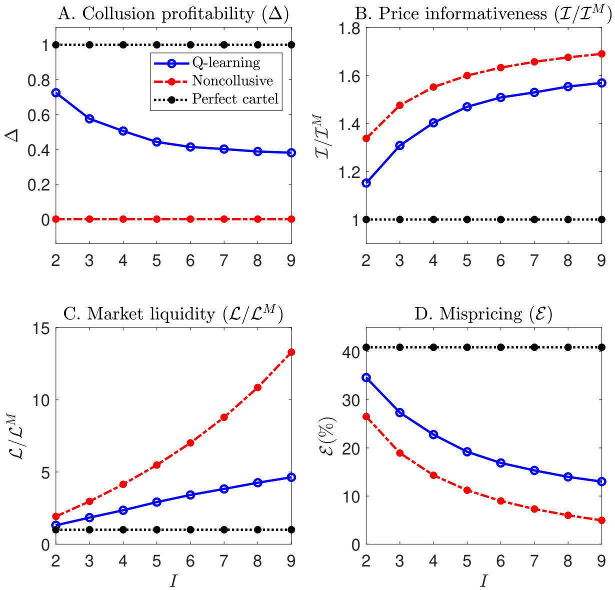
No Folk Theorem: Over-pruning bias ($\sigma_u/\sigma_v = 10^2$)



Concentration: Price-trigger strategies ($\sigma_u/\sigma_v = 10^{-1}$)



Concentration: Over-pruning bias ($\sigma_u/\sigma_v = 10^2$)



Conclusion

This paper studies the "psychology" of AI traders

- Theory of learning in games is useful for understanding AI equilibrium

"AI collusion" emerges without communication or intended codes

- Through price-trigger strategies (artificial "intelligence")
- Through over-pruning bias in learning (artificial "stupidity")

"AI collusion" undermines market efficiency

- Reduced market liquidity
- Diminished price informativeness
- Increased mispricing

Policy innovations (future research)

- Rethink the market manipulation law
- Limit state variable inclusion for RL algorithms
- Deploy AI algos on the platform to counteract "AI collusion"
- Prevent AI concentration and homogenization