Algorithmic Collusion at Test Time: A Meta-game Design and Evaluation

Yuhong Luo
Rutgers University
New Brunswick, USA
y.luo@rutgers.edu

Daniel Schoepflin
DIMACS, Rutgers University
New Brunswick, USA
daniel.r.schoepflin@gmail.com

Xintong Wang
Rutgers University
New Brunswick, USA
xintong.wang@rutgers.edu

ABSTRACT

The threat of algorithmic collusion, and whether it merits regulatory intervention, remains debated, as existing evaluations of its emergence often rely on long learning horizons, assumptions about counterparty rationality in adopting collusive strategies, and symmetry in hyperparameters and economic settings among players. To study collusion risk, we introduce a meta-game design for analyzing algorithmic behavior under test-time constraints. We model agents as possessing pretrained policies with distinct strategic characteristics (e.g., competitive, naively cooperative, robustly collusive), and formulate the problem as selecting a meta-strategy that combines a pretrained, initial policy with an in-game adaptation rule. We seek to examine whether collusion can emerge under rational choices and how agents co-adapt toward cooperation or competition. To this end, we sample normal-form empirical games over meta-strategy profiles, compute relevant game statistics (e.g., payoffs against individuals and regret against an equilibrium mixture of opponents), and construct empirical best-response graphs to uncover strategic relationships. We evaluate reinforcement-learning, UCB, and LLM-based strategies in repeated pricing games under symmetric and asymmetric cost settings, and present findings on the feasibility of algorithmic collusion and the effectiveness of pricing strategies in test-time environments.[1]

KEYWORDS

Algorithmic collusion; repeated price competition; meta-game; test-time adaptation

ACM Reference Format:
Yuhong Luo, Daniel Schoepflin, and Xintong Wang. 2026. Algorithmic Collusion at Test Time: A Meta-game Design and Evaluation. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 28 pages. https://doi.org/10.65109/SVWG7670

1 INTRODUCTION

The use of algorithms to automate economic decisions—such as pricing, bidding, and bargaining—has become increasingly prevalent. Rather than following static rules, modern systems leverage optimization techniques (e.g., reinforcement learning) or AI models with reasoning capabilities (e.g., large language models) to learn from their environment, anticipate the behavior of other participants, and adapt their strategies to improve outcomes.

This growing autonomy introduces new, critical risks including the undesired and uncommunicated cooperation among algorithms, commonly known as *algorithmic collusion*. Recognized as a major concern in the era of advanced AI [27], algorithmic collusion poses unique challenges beyond human collusion [4, 16], as algorithms may learn to coordinate without explicit communication or intent [44]. Prior work demonstrates that collusive behavior can arise in simulated pricing and auction environments using common algorithms such as Q-learning [14] and large language models (LLMs) [23]. However, whether such behavior persists under realistic conditions and deployments remains unclear. For example, Calvano et al. [14] document convergence to collusion between two Q-learning agents after around 1.5 million interaction rounds, reflecting a prohibitively long learning horizon with significant early-stage exploration costs.

A more practical evaluation considers test-time interactions, where pretrained policies are randomly rematched and allowed to interact for a limited number of rounds [14, 20]. These studies show that while collusion may fade at first, it can re-emerge after roughly 40,000 rounds of play. Importantly, prior analyses largely assume symmetric algorithmic configurations (e.g., identical hyperparameters) and leave open a central question: whether collusion constitutes a rational, stable outcome when agents strategically behave under test-time constraints.

In this work, we seek to evaluate the viability and effectiveness of algorithmic collusion in "test-time" settings, where an agent must adapt within a limited number of interactions to opponents whose policies and economic settings (e.g., cost or quality) may differ from those encountered during training. To this end, we propose a meta-game framework. Different learning algorithms (e.g., Q-learning, no-regret methods, and LLMs) are first used at training time to generate candidate *initial policies* for deployment. These policies are then grouped into strategic categories based on performance metrics, including their propensity to collude with a training-time partner and their robustness against a best-response opponent. At test time, a *strategy* is formed by pairing an initial policy sampled from a selected category with an adaptation rule (e.g., a learning rate) drawn from a set of feasible update procedures for repeated play. A *meta-strategy* governs both the choice of initial policy category and the manner in which the policy adapts during interaction.

We are interested in analyzing these competing meta-strategies by modeling their interactions as a game, referred to as the *meta-game*. By sampling strategies for each player together with initial states, we generate meta-game instances from which evaluation

---

[1] The full paper with appendix is available at: https://arxiv.org/abs/2602.17203. The source code can be found at: https://github.com/chailab-rutgers/CollusionMetagame.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/SVWG7670

metrics can be computed and game-theoretic analysis can be further conducted. Through this framework, we provide a statistical characterization of meta-strategy performance and address the central question: Can algorithmic collusion emerge within a limited time horizon under rational strategic choices?

Our contribution. We propose metrics to characterize and evaluate a (pretrained) policy and introduce a meta-game framework to assess algorithmic collusion at test time through interactions among meta-strategies. We conduct extensive experiments using a diverse set of algorithms (including Q-learning, UCB, and LLMs) to generate initial policies for repeated pricing games. Our results provide empirical evidence on meta-strategy performance, and offer insights into the conditions under which algorithmic collusion may emerge and persist. We highlight several key findings below:

- Q-learning can produce pretrained policies that *robustly collude* with their best-response counterparts, whereas UCB-based policies, despite colluding with their training-time partners, are often exploitable by best-response opponents.
- Under symmetric cost settings, each algorithm we evaluate—Q-learning, UCB, LLM—admits at least one pure or mixed Nash equilibrium among meta-strategies that leads to a collusive outcome, indicating that collusion can arise from rational strategic choices.
- For Q-learning meta-strategies, collusion diminishes under shorter interaction horizons and pessimistic Q-value initialization, which can be interpreted as reflecting an agent's prior belief that its counterparty is less likely to collude. Notably, in contrast to prior studies based on symmetric algorithmic setups that observe sustained collusion in asymmetric cost settings [14], our findings reveal that rational strategy selection substantially suppresses collusion under asymmetry.
- While UCB meta-strategies exhibit stronger collusion than Q-learning overall, Q-learning with random initialization can best respond to most UCB meta-strategies, calling into question UCB's competitiveness under test-time conditions.
- LLM-based agents can demonstrate adaptive behavior guided by pre-game history: policies that exhibit greater collusion during pre-game interactions tend to sustain or re-establish collusion even after episodes of exploitation at test time.

1.1 Related Work

Algorithmic collusion. Collusion can arise as an equilibrium among rational players in infinitely repeated games. Building on the Folk Theorem, prior work shows that cooperation can be sustained when algorithms embed credible punishment for deviations [3, 28, 36, 51]. More recent studies find that collusive equilibria may also emerge without explicit threat mechanisms, as shown in a Stackelberg framework [6], and examine how algorithmic (non-)collusion can be detected and regulated through data audits [29, 30].

Collusive behavior has been documented in e-commerce and dynamic pricing settings [7, 12, 13, 22, 43]. Following Calvano et al. [14], a growing body of literature investigates algorithmic collusion in the laboratory across settings ranging from stylized games, such as the repeated Prisoner's Dilemma [10, 42], to more realistic auctions [9]. These studies also consider different algorithms, including reinforcement learning [31, 35], bounded-regret methods [28], and LLMs [5, 23]. Several follow-up studies suggest that prior findings may overstate the threat of algorithmic collusion, as they rely on restrictive modeling assumptions [1]. Collusion typically emerges after long training horizons—on the order of millions of repeated-game rounds—and tends to depend on symmetric algorithmic configurations, suggesting implicit coordination or prior communication [37]. Moreover, the algorithmic configurations that give rise to collusion may be irrational in practice, being outperformed by more robust or commonly used alternatives [2, 11]. Thus, the emergence of algorithmic *tacit* collusion among rational agents under realistic conditions and finite interaction horizons remains an open question.

Strategy selection, adaptation, and evaluation. Our work focuses on strategy performance at deployment, resembling a test-time or tournament environment. Prior research has examined strategy selection and evaluation through meta-game frameworks [34, 38] and empirical game-theoretic analysis (EGTA) [59], applied to domains such as trading agent competitions [21, 45, 47], financial markets [53, 55–57], supply chain management [33], negotiation [38], and auctions [52]. Related to our setting is Carissimo et al. [15], who model collusion as coordination in Q-learning hyperparameters (e.g., discount and exploration factors) during training. Our study differs by focusing on test-time adaptation among pretrained agents with heterogeneous initializations, distinguishing deployment behavior from training dynamics.

Related online learning literature also examines *adaptability* and *non-exploitability* in repeated games [17–19], and develops approximate best-response methods for equilibrium analysis in complex games such as poker and Go [39, 40, 50]. These insights inform our design of metrics for characterizing pretrained policies and their adaptation strategies.

2 PRELIMINARIES

We consider an n-player simultaneous-move repeated game progressing from time 0 to an end time unknown to the players. At each round t, a player j commits to an *action* (e.g., price) $p_{j,t} \in \mathcal{P}$, where $\mathcal{P}$ denotes a discrete action space.

The game admits a *Stage-Game Nash Equilibrium* (SGNE) in which no player can profitably deviate in a single round, given that stage games are independent across periods. Let r_j^N denote player j's *competitive payoff*, $\bar{r}^N$ the average competitive payoff, and p_j^N the *competitive action* of player j under the SGNE. In pricing settings with homogeneous goods and symmetric costs, this corresponds to the *Bertrand Nash equilibrium*.

Players may nonetheless achieve higher payoffs through coordinated behavior that avoids the SGNE. The optimal joint outcome for all players in a stage game can be derived by maximizing the joint payoffs; we refer to this outcome as *the monopoly* which characterizes full collusion. In pricing contexts, such outcomes are detrimental to consumers. Let r_j^M denote player j's *monopoly payoff*, $\bar{r}^M$ the average monopoly payoff, and p_j^M the associated *monopoly action* of player j.

The repeated pricing game can be modeled as an extensive-form game whose central solution concept is the *subgame perfect equilibrium* (SPE)—a strategy profile that constitutes a Nash equilibrium in every subgame. According to the Folk Theorem [24], many SPEs exist, with each supported by *credible threats*, i.e., responses that are rational for the player making them [48]. By contrast, non-credible threats cannot sustain an SPE, and monopoly outcomes often depend on such threats. Indeed, in our experiments, we find no evidence of credible threats among pretrained policy pairs, suggesting that realistic collusion may not rely on them.

Following Calvano et al. [14], we define the Collusion Index (CoI) as a measure of collusion: when CoI = 0%, players are fully competitive by playing SGNE, whereas CoI = 100% corresponds to full collusion at the monopoly outcome.

Definition 2.1 (Collusion Index (CoI)).

$$CoI := \frac{\bar{r} - \bar{r}^N}{\bar{r}^M - \bar{r}^N}, \text{ where } \bar{r} \text{ is the average per-player payoff.}$$

We define a state of the game $s \in \mathcal{S}$ as actions (i.e., prices) of all players in one round, ordered by the player IDs. In this work, we focus on two-player repeated games. We let S_t be the random variable of the state of the game at timestamp t. A policy $\pi_{j,t} \in \Pi$ for player j at time t maps the state $S_t = s$ to a probability distribution over actions. Given a discount factor γ and a fixed policy profile (π_j, π_{-j}), let $V^{\pi_j|\pi_{-j}}(s)$ denote the expected utility of player j starting from state s (formally defined in Appendix A.1).

3 META-GAME DESIGN FOR ALGORITHMIC COLLUSION

This section introduces our meta-game design to reason about how players select and adapt their policies in "test-time" environments with limited interactions, and assess whether rational choices can lead to collusive outcomes.

3.1 Initial Policy, Strategy, and Meta-strategy

3.1.1 *Initial Policy.*

As in many multi-agent domains, agents begin play with a pretrained policy derived from prior experience, we consider agents that start with an initial policy generated via a pretraining phase. We model this by applying a learning algorithm $\mathcal{A}_\theta$ and a set of random seeds $\mathcal{K}$ to a game of interest $\mathcal{G}$ to generate a set of initial policy profiles Π. We assume each pretraining process, i.e., a stochastic procedure that produces a policy profile $\pi_\kappa = \{\pi_1^\kappa, ..., \pi_N^\kappa\} = \mathcal{A}_\theta(\mathcal{G}, \kappa)$, is subject to no time constraint or cost of learning.

3.1.2 *Strategy.*

Without adaptation, however, pretrained policies can fail against unfamiliar opponents, resulting in unexpected state transitions and suboptimal outcomes at test time.[2] Achieving strong performance thus requires an update procedure to adapt the initial policy to the specific opponent during play. We define an agent's strategy for the game at test time $\mathcal{G}^*$ as the combination of a (pretrained) initial policy and an update procedure. This strategy is evaluated over a shorter interaction horizon, where learning incurs costs and performance at each round counts.

Rather than directly modifying the policy, we consider strategies that operate on an underlying *internal representation* that encodes the policy. We denote this representation by $Z_{j,t}$ for player j at time t, and it is updated after each round. For example, in Q-learning, the internal representation consists of Q-values, from which the agent selects the action with the highest value.

Formally, a strategy is specified by two functions, typically determined by the learning algorithm and its hyperparameters:

- A *decoding function* ϕ which maps the internal representation to a policy, $\phi(Z_{j,t}) = \pi_{j,t}$, and
- An *update function* ω which updates the representation based on the latest experience (i.e., the state), $\omega(Z_{j,t}, S_{j,t}) = Z_{j,t+1}$.

At any time t, a player's strategy is fully specified by its current representation $Z_{j,t}$ and its decoding and update functions (ϕ_j, ω_j). In Section 4, we will illustrate how different algorithms, including Q-learning, UCB, and LLMs, instantiate these components.

3.1.3 *Meta-strategy.*

Pretraining can generate a large set of initial policies, each of which may be paired with infinitely many update procedures (e.g., varying learning rates). As a result, the induced strategy space becomes prohibitively large, rendering direct and systematic analysis intractable. To manage this complexity, we group initial policies according to their performance along two key strategic dimensions: their ability to achieve cooperation (i.e., how effectively a policy learns to cooperate with a specific partner) and their ability to avoid exploitation (i.e., how robustly the policy performs against a best-response opponent). We begin by introducing metrics to evaluate these dimensions.

Definition 3.1 (Paired Cooperativeness (PC)). *Let π_j and π_{-j} denote a pair of policies. The paired cooperativeness between π_j and π_{-j} is defined as the respective mean state values when interacting with one another, i.e.,*

$$PC(\pi_j, \pi_{-j}) = \left(\overline{V}^{\pi_j|\pi_{-j}}, \overline{V}^{\pi_{-j}|\pi_j}\right), \text{ where } \overline{V}^{\pi_x|\pi_y} = \frac{\sum_{s\in\mathcal{S}} V^{\pi_x|\pi_y}(s)}{|\mathcal{S}|}.$$

Paired Cooperativeness (PC) measures the expected utility of each policy in a pair when the initial state is drawn uniformly at random. Intuitively, jointly pretrained policy pairs tend to exhibit high PC values, with positively correlated mean state values, as their trajectories are trained to converge to collusive outcomes from any initial state. In contrast, independently pretrained policies paired at test time are less likely to reach collusive absorbing states, leading to lower PC values and increasing the likelihood of one policy exploiting the other.

With PC capturing how well two policies sustain cooperation, we further introduce *cooperative robustness* (CR), which evaluates a policy's performance against its best response *and* its tendency to cooperate with that best response.

Definition 3.2 (Cooperative Robustness (CR)). *Let π_b denote the worst-case best-response policy to π_j.*

$$\pi_b = \arg\min_{\pi\in BR(\pi_j)} \overline{V}^{\pi_j|\pi}, \text{ where } BR(\pi_j) := \arg\max_{\pi\in\Pi} V^{\pi|\pi_j}(s), \forall s \in \mathcal{S}.$$

The best responses can be obtained via value iteration [49]. The cooperative robustness between π_j and π_b is defined as their respective mean state values when interacting with each other:

$$CR(\pi_j, \pi_b) = \left(\overline{V}^{\pi_j|\pi_b}, \overline{V}^{\pi_b|\pi_j}\right).$$

Intuitively, the relative magnitudes of $\overline{V}^{\pi_j|\pi_b}$ and $\overline{V}^{\pi_b|\pi_j}$ characterize the strategic nature of the interaction between the two policies. When these values are highly asymmetric, it indicates an exploitative relationship, typically with π_b exploiting π_j, which

---

[2] Indeed, in experiments, we find that two independently pretrained Q-learning agents, when paired against each other without adaptation, can yield payoffs lower than competitive pricing (see Fig. 4 in the appendix).

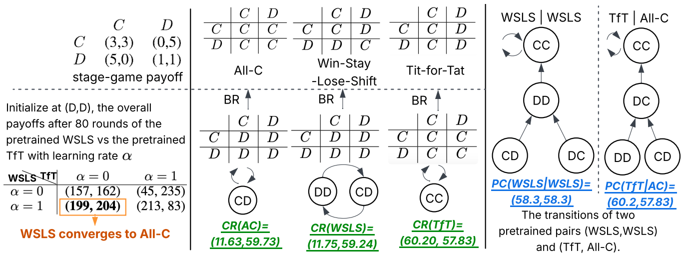

Figure 1: A toy meta-game for repeated Prisoner’s Dilemma on canonical strategies. The NE of this meta-game leads to cooperation. See Example 3.1 for more discussion.

reflects π_j’s lack of robustness. When both values are low, the interaction tends to settle in a competitive absorbing state, indicating robustness but limited cooperative gains. In contrast, when both values are high and comparable, the two policies exhibit robust cooperation, achieving mutually beneficial outcomes while remaining stable against unilateral deviations.[3]

We characterize pretrained initial policies along these two strategic dimensions and group them into distinct categories. Policies within each category are then paired with a discrete set of update procedures. We define a *meta-strategy* as a family of strategies formed by combining an initial policy category—defined by specific strategic attributes (e.g., in PC and CR)—with a corresponding update rule. Practically, each meta-strategy specifies how an agent *selects and adapts* an initial policy from the pool of candidates generated during pretraining.

Below we provide a motivating example based on canonical strategies in the repeated Prisoner’s Dilemma, coupled with simple update rules, as illustrated in Fig. 1.

Example 3.1 (A toy meta-game for repeated Prisoner’s Dilemma). *Consider the following initial policies for repeated Prisoner’s Dilemma: Win-Stay, Lose-Shift (`WSLS`), Tit-for-Tat (`TfT`), and Always-Cooperate (`AC`). Suppose they are obtained from two pretrained policy pairs that successfully learned to cooperate, i.e., (`WSLS`, `WSLS`) and (`TfT`, `AC`).*

We categorize these policies. All policies achieve high PC with their respective pretrained partners due to cooperative behavior. However, `WSLS` and `AC` both yield low mean state values against their best responses, resulting in highly asymmetric CR, as they can be exploited by a best-response policy (e.g., always Defect). By contrast, `TfT` attains high CR, as its best response is to always cooperate.

Consider four meta-strategies constructed by pairing high-PC–low-CR and high-PC–high-CR initial policies with either slow or fast in-game update rules. Suppose `WSLS` and `TfT` are sampled as representative initial policies from the two categories and are paired with low and high learning rates. Fig. 1 presents one part of the empirical meta-game defined over this restricted, sampled strategy space. Simulated profile payoffs indicate a meta-game equilibrium in which `WSLS` updates its policy and `TfT` remains unchanged.[4]

We are interested in sampling and aggregating many such payoffs to construct empirical meta-games and analyze the relative performance of meta-strategies. For instance, in a different meta-game, `AC` could be sampled as an alternative initial policy from the high-PC–low-CR category.

3.2 Empirical Game-theoretic Analysis

Our meta-game evaluation leverages empirical game-theoretic analysis (EGTA), a methodology for analyzing complex game situations and strategic interactions through agent-based simulation [58, 59]. EGTA constructs an empirical normal-form game over a selected set of strategies using payoff data generated from simulations of the underlying environment.

We apply EGTA to a repeated two-player, general-sum game (e.g., repeated pricing or auction) to investigate algorithmic collusion. Given a set of M meta-strategies, our goal is to evaluate their relative performance in the test-time, base game $\mathcal{G}^*$. Let $\hat{\psi}^m$ denote a strategy sampled from meta-strategy $\mathcal{M}^m$ and let $\hat{\Psi} = \{\hat{\psi}^1, \dots, \hat{\psi}^M\}$ represent the set of all sampled strategies. We then construct an empirical meta-game $\mathcal{MG}(\hat{\Psi})$ over this sampled strategy space.

When the base game $\mathcal{G}^*$ is symmetric (e.g., agents have the same cost and quality parameters), we estimate the meta-game payoff function by simulating two-player strategy profiles over $\hat{\Psi}$ to play $\mathcal{G}^*$ (due to symmetry, the assignment of strategies to players does not matter).[5] The empirical meta-game payoff matrix is estimated by repeatedly simulating $\mathcal{G}^*$ for each two-player strategy profile over different initial states sampled from $\mathcal{S}$ across runs. When the base game $\mathcal{G}^*$ is asymmetric, agents are pretrained on different parameters and may possess distinct strategies. We therefore sample $\hat{\Psi}_1 = \{\hat{\psi}_1^1, \dots, \hat{\psi}_1^M\}$ and $\hat{\Psi}_2 = \{\hat{\psi}_2^1, \dots, \hat{\psi}_2^M\}$ from their respective meta-strategy sets to construct the empirical game.

3.3 A Meta-game Evaluation Framework

Given a two-player, general-sum repeated game $\mathcal{G}^*$ and meta-strategies $\{\mathcal{M}^1, \dots, \mathcal{M}^M\}$, we construct empirical meta-games via the following sampling and evaluation procedure in Algorithm 1.

Algorithm 1: Meta-game evaluation procedure

```
Input  : Meta-strategy set $\Psi$, base game $\mathcal{G}^*$, number of
         meta iterations $N_{\text{meta}}$, number of base game runs
         $N_{\text{base}}$
Output : Empirical meta-games and game-analysis stats $X$
repeat
    Construct a strategy set $\hat{\Psi} = \{\hat{\psi}^1, \dots, \hat{\psi}^M\}$ by uniformly
      sampling one strategy from each meta-strategy in $\Psi$.
    repeat
        Sample an initial state $s_0 \sim \mathcal{S}$ for the base game $\mathcal{G}^*$.
        Estimate the empirical $\mathcal{MG}(\hat{\Psi})$ by simulating
          two-player profiles as described in Section 3.2, and
          record the resulting profile payoffs.
        Compute the desired statistics $X$ from $\mathcal{MG}(\hat{\Psi})$.
    until $N_{base}$;
until $N_{meta}$;
```

---

[3] See Fig. 5 in the appendix for examples of Q-learning policies in different strategic categories.

[4] Q-learning is adopted for both pretraining and test-time adaptation with $\gamma = 0.95$. We normalize initial Q-values following the procedure described in Appendix C.3 and set exploration $\varepsilon = 0$. We evaluate strategies at $t = 80$, when both strategies converged.

[5] To prevent collusion arising from both players using an identical or jointly pretrained policy, if the same policy or its pretrained pair has been drawn, we redraw an instance.

Game-analysis statistics. We construct *weighted best-response graphs* using the payoff matrices of meta-games generated by Algorithm 1. For each meta-game, the *best-response score* from strategy u to v is defined as the ratio of u's average payoff against v to the highest average payoff achievable by any strategy against v. The best-response scores across all meta-games are aggregated to determine the edge weights. If $\mathcal{G}^*$ is asymmetric, we maintain two directed best-response graphs, one for each player role.

We evaluate the *uniform score* of a meta-strategy, i.e., the expected payoff of a meta-strategy when the opponent is drawn from a uniform distribution over all meta-strategies. As uniform scores represent a naive belief of the opponent distribution, we also adopt an alternative metric, NE-regret, as proposed by Jordan et al. [33].

Definition 3.3 (NE-Regret). *Suppose σ^* is a symmetric mixed-strategy NE of $\mathcal{MG}(\Psi)$. The NE-regret of a pure meta-strategy $\mathcal{M}_j$ is defined as the difference between the expected payoff of player j of the equilibrium profile (σ^*, σ^*) and that of the mixed-strategy profile $(\mathcal{M}_j, \sigma^*)$.*

A high NE-regret can be due to an inability to cooperate or susceptibility to exploitation. As a meta-game may admit multiple MSNEs with varying degrees of cooperativeness, we follow Balduzzi et al. [8]'s *Nash averaging* and report NE-regret with respect to the max-entropy NE, which captures the broadest set of meta-strategies.

4 EVALUATING ALGORITHMIC COLLUSION IN REPEATED PRICING GAMES

We first introduce the repeated pricing game, and then conduct the meta-game evaluation on three common pricing algorithms: Q-learning, UCB, and LLMs.

4.1 Economic Environment

We consider the canonical repeated pricing game in which firms (i.e., agents) act simultaneously and condition their actions on history. For the stage game, we adopt a simple model of price competition with *logit demand*, which has been widely applied [12, 14, 41]. The details of the logit demand model is in Appendix B.1.

We focus on a two-agent repeated pricing game as our base game $\mathcal{G}^*$. Each agent has private information about its marginal cost c_j and product quality a_j, both of which remain fixed throughout $\mathcal{G}^*$.[6] An agent can observe its demand $d_{j,t}$ and both firms' prices. The game state of period t is $S_t \coloneqq (p_{j,t}, p_{-j,t})$. The profit (i.e., payoff) for agent j at period t is $r_j(S_t) \coloneqq (p_{j,t} - c_j) \cdot d_{j,t}(S_t)$. Following Calvano et al. [14], prices are discretized and equally spaced, forming the agent's action space. We consider the discretization levels of 4 and 15. For a discretization of N_{discrete}, let the step size be $\xi = \frac{p_j^M - p_j^N}{N_{\text{discrete}} - 2}$ where p_j^N is the competitive price, p_j^M the monopoly price. The action space is $\mathcal{P}_j \coloneqq \left\{ p_j^N + (k-1)\xi \mid k \in \{0, 1, \ldots, N_{\text{discrete}} - 1\} \right\}$.

4.1.1 Meta-strategies.

The base game $\mathcal{G}^*$ takes as input a pair of pricing strategies and returns the corresponding payoffs for each agent. We can evaluate cumulative payoffs at any point during $\mathcal{G}^*$, capturing the effect of varying test-time horizon and the cost of learning.

We use three algorithms—Q-learning, UCB, and LLM—to generate pretrained initial policies. Formally, each algorithm together with its associated hyperparameters defines a stochastic procedure that produces a policy profile $(\pi_j, \pi'_j) = \mathcal{A}_\theta(\mathcal{G}, \kappa)$, given a random seed κ. As described in Section 3.1, we classify pretrained policies according to their paired cooperativeness (PC; Def. 3.1) with their pretraining partners and their cooperative robustness (CR; Def. 3.2).[7] From the pretrained pool, we select three representative categories (illustrated in Appendix Fig. 5):

1. Less colluding (LC): Policies in the bottom third of $\overline{V}^{\pi_j|\pi'_j}$ and the middle third of $\overline{V}^{\pi_j|\pi_b}$, representing policies that achieve little collusion with their pretrained partner upon convergence. With a price space discretized to four levels, these typically correspond to competitive pricing policies.
2. Colluding (C): Policies in the top third of $\overline{V}^{\pi_j|\pi'_j}$ and the bottom third of $\overline{V}^{\pi_j|\pi_b}$, indicating collusion with pretrained partners but vulnerability to best-response exploitation.
3. Robust colluding (RC): Policies in the top third for both $\overline{V}^{\pi_j|\pi'_j}$ and $\overline{V}^{\pi_j|\pi_b}$.

Tertiles are computed within each pretraining algorithm. We also consider a baseline category of randomly initialized (RD) policies. Each initial policy category is paired with a set of algorithm-specific update rules spanning a range of adaptation speeds, from fast to slow. We discuss Q-learning in Sec. 4.2, UCB in Appendix D.1 and LLM in Sec. 4.3. For all meta-game evaluations, additional results including the payoff matrices and the best-response graphs are referenced in Appendix C.8 for Q-learning, Appendix D.3 for UCB, and Appendix E.3 for LLM.

4.2 Tabular Q-learning for Pricing

Tabular Q-learning is a simple yet effective algorithm widely adopted in pricing settings [14, 54]. The corresponding decoding and update rules are in Appendix C.2.

Pretraining. Our pretraining procedure follows Calvano et al. [14]. We randomize the learning rate α, exploration rate ε, and its decay δ, while fixing the discount factor $\gamma = 0.95$. Cost and quality parameters are kept identical to agents' test-time values. All pretrained pairs use symmetric settings and play until their policies converge. We run 500 pretraining games with different random seeds, yielding 1,000 pretrained policies.

Initialization of Q-values. Initial Q-values influence exploration and are a key determinant of test-time adaptation [32, 46]. Since pretrained Q-values can vary widely across state–action pairs due to randomness and exploration during training, we rescale them to standardize their maximum value while preserving the induced policy and the ordering of actions within each state. This rescaling allows for comparisons across policies by controlling for differences in Q-value magnitudes. We introduce a rescaling factor f that interpolates between optimistic ($f = 1$) and pessimistic ($f = 0$) initializations, corresponding to converged Q-values at collusive and competitive absorbing states respectively. The rescaling procedure is provided in Appendix C.3. Section 4.2.3 presents meta-game evaluations on meta-strategies of different scaling factors.

---

[6] We consider symmetric product qualities $a_j = a_{-j} = 2$, while allowing for cost asymmetries in certain settings.

[7] Given the strong correlation between the two dimensions of PC and CR (see Appendix Fig. 6), we base the categorization on a single dimension.

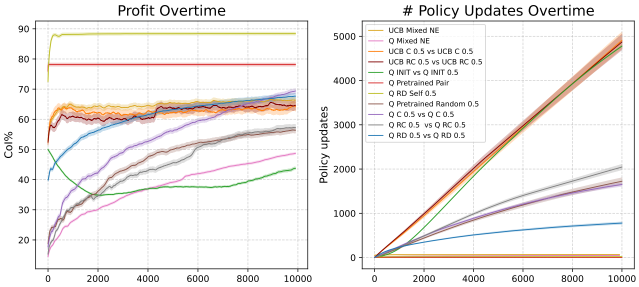

Figure 2: Running averages of CoI over 100 rounds (Left) and accumulated policy update counts for strategy pairs over 10,000 rounds (Right). Shaded regions indicate 95% confidence intervals. Each curve represents the mean over 20 strategy pairs and 100 random initial states. INIT denotes Q-learning agents trained from scratch, with Q-values initialized to those corresponding to opponents playing uniformly random pricing strategies. All strategies except INIT and pretrained pairs use $\alpha = 0.5$ and $f = 1$.

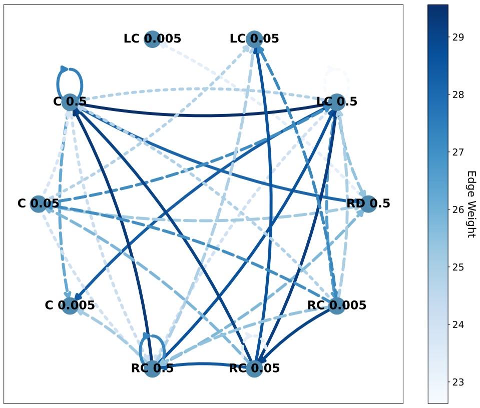

Figure 3: The best-response graph for Q-learning with $c_1 = c_2 = 1$, evaluated at $t = 10,000$. The edge weights correspond to the sum of best-response scores across all meta-games as discussed in Sec. 3.3.

Game settings. For the base repeated pricing games, we examine two symmetric cost settings ($c_1 = c_2 = 1$ and $c_1 = c_2 = 0.8$) and one asymmetric setting ($c_1 = 1, c_2 = 0.8$), with $N_{\text{discrete}} = 15$, $N_{\text{base}} = 100$ and $N_{\text{meta}} = 40$. We evaluate a set of ten meta-strategies (Table 1), spanning combinations of initial policy categories and learning rates $\alpha \in \{0.5, 0.05, 0.005\}$. Other meta-strategies—such as random initialization with smaller learning rates or pretrained policies with larger ones—were excluded based on preliminary experiments, as they were strictly dominated. We further explore the rescaling factor as a meta-strategy parameter for $f = \{1, 0.5, 0\}$.

4.2.1 *Symmetric Costs.*

Table 1 and Fig. 3 present our main findings for symmetric base games, evaluated at $t = 10,000$.[9] We identify a PSNE at (C, 0.5) and a max-entropy MSNE involving three meta-strategies: (C, 0.5), (RC, 0.05), and (RC, 0.005).

Fig. 2 reports the average CoI performance of paired strategies sampled from the respective meta-strategies over 2,000 runs. With symmetric costs of one, collusion emerges as a rational adaptive outcome, driven by agents' strategic choices among the available meta-strategies: strategies from the PSNE achieve approximately 70% CoI, while those from the max-entropy MSNE reach around 50%. Furthermore, we highlight that when jointly pretrained or identically initialized random policies are paired, CoI can exceed 80% with minimal or no adaptation, indicating that prior coordination can greatly amplify collusive behavior. These adaptation patterns may serve as useful signals for identifying potential prior coordination.

Fig. 3 illustrates the strategic relationship among meta-strategies. Within the RC family, strategies with smaller learning rates tend to dominate those with larger ones, suggesting the value of preserving a robust initial policy at test time. In contrast, C meta-strategies benefit from larger learning rates, which enable faster adaptation and reduce vulnerability to exploitation. We also note that while (RD, 0.5) achieves moderate collusion (around 65% CoI as shown in Fig. 2) when matched against itself, it is easily exploited by most pretrained strategies, rendering it strategically dominated.

The discussion for how shorter evaluation horizons $t = 3,000$ (Table 8), lower symmetric costs (Table 1), and different Q-value rescaling factors (Table 2) affect the rational selection of meta-strategies is deferred to Appendix C.5 and their payoff matrices and BR graphs to Appendix C.8.

4.2.2 *Asymmetric Costs.*

Table 1 presents the main results of the meta-game evaluation performed on repeated pricing with asymmetric costs ($c_1 = 1, c_2 = 0.8$). The corresponding payoff matrices and best-response graphs are provided in Appendix C.8. The max-entropy MSNE reveals that the low-cost agent chooses strategies from the LC family and the high-cost agent selects from the C and RC families. The low-cost agent enjoys a natural cost advantage and thus adopts LC strategies to play competitively with minimal adaptation. In response, the high-cost player selects RC strategies to remain robust against exploitation while attempting to sustain some level of collusion. Overall, collusion no longer persists, as the low-cost player has stronger incentives and greater opportunities to deviate and exploit.

We note that our findings here differ from those of Calvano et al. [14], who report sustained high collusion (75.9% CoI) under even more asymmetric scenarios ($\frac{c_1}{c_2} = 2$) with symmetric algorithmic configurations. Our finding suggests that incorporating rationality into strategy selection and deployment decisions may suppress collusive behavior in asymmetric settings.

Our meta-game analysis assumes that each player knows the opponent's cost type. In practice, agents may hold beliefs over the opponent's type, influencing their strategic choices. Extending the framework to a Bayesian Nash equilibrium would better capture behavior under incomplete information.

4.2.3 *Asymmetric Rescaling: A Meta-game on Pessimistic vs. Optimistic Initialization.*

We construct a meta-game where the set of meta-strategies includes a mixture of rescaling factors, $f \in$

---

[8] Uniform scores in all tables are converted to the CoI scale based on the following transformation: $x \rightarrow \frac{x-\bar{r}^N}{\bar{r}^M-\bar{r}^N} \times 100\%$, where x is the expected payoff of a meta-strategy in a stage game when the opponent's strategy is drawn from a uniform distribution over all meta-strategies. We denote $\bar{r}^N$ as the average competitive payoff and $\bar{r}^M$ the average monopoly payoff (Sec. 2).

[9] Based on Amazon's 15-minute pricing updates, $t = 3,000$ is approximately one month, and $t = 10,000$ approximates three months. https://sell.amazon.com/tools/automate-pricing

Table 1: Max-entropy MSNE, NE-regret ($\times 10^{-3}$), and the uniform score (converted to the CoI scale) evaluated at $t = 10,000$ for Q-learning with optimistic initialization $f = 1$. These metrics are defined in Sec. 3.3.[8]The best-performing strategy is highlighted in bold. The second-best is underlined. The ± symbol denotes the 95% confidence interval.

$t = 10,000$, $f = 1$		RD 0.5	LC 0.5	LC 0.05	LC 0.005	C 0.5	C 0.05	C 0.005	RC 0.5	RC 0.05	RC 0.005
PSNE	$c_1 = c_2 = 1$	-	-	-	-	✓	-	-	-	-	-
MSNE	$c_1 = c_2 = 1$	0.00	0.00	0.00	0.00	0.49	0.00	0.00	0.10	0.28	0.13
NE-Regret ($\times 10^{-3}$)	$c_1 = c_2 = 1$	8.10 ± 0.41	1.05 ± 0.43	7.09 ± 0.56	12.76 ± 0.68	**0.00 ± 0.37**	5.60 ± 0.53	10.56 ± 0.57	**0.00 ± 0.49**	**0.00 ± 0.51**	**0.00 ± 0.59**
Uniform Score	$c_1 = c_2 = 1$	35.21 ± 0.32	39.44 ± 0.29	33.08 ± 0.35	28.01 ± 0.43	**40.96 ± 0.22**	36.10 ± 0.22	32.64 ± 0.32	39.66 ± 0.25	37.90 ± 0.24	37.52 ± 0.29
PSNE	$c_1 = c_2 = 0.8$	-	-	-	✓	-	-	-	-	-	-
MSNE	$c_1 = c_2 = 0.8$	0.00	0.00	0.00	0.23	0.17	0.00	0.08	0.00	0.00	0.53
NE-Regret ($\times 10^{-3}$)	$c_1 = c_2 = 0.8$	7.72 ± 0.53	2.94 ± 0.64	4.34 ± 0.54	**0.00 ± 0.50**	**0.00 ± 0.59**	6.78 ± 0.43	**0.00 ± 0.41**	7.97 ± 0.69	11.20 ± 0.47	**0.00 ± 0.42**
Uniform Score	$c_1 = c_2 = 0.8$	29.26 ± 0.25	32.84 ± 0.26	32.06 ± 0.21	32.01 ± 0.21	**35.14 ± 0.19**	31.69 ± 0.18	30.15 ± 0.22	32.03 ± 0.25	33.21 ± 0.20	34.83 ± 0.20
MSNE	$c_1 = 1.0$	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.00	0.00	0.76
	$c_2 = 0.8$	0.00	0.09	0.00	0.91	0.00	0.00	0.00	0.00	0.00	0.00
NE-Regret ($\times 10^{-3}$)	$c_1 = 1.0$	14.25 ± 0.85	13.94 ± 0.92	7.69 ± 0.77	3.38 ± 0.56	18.58 ± 0.75	8.61 ± 0.55	**0.00 ± 0.50**	13.88 ± 0.84	8.12 ± 0.48	**0.00 ± 0.36**
	$c_2 = 0.8$	18.47 ± 0.67	**0.00 ± 0.75**	0.26 ± 0.54	**0.00 ± 0.50**	14.90 ± 0.57	25.48 ± 0.38	24.21 ± 0.45	1.88 ± 0.75	17.65 ± 0.58	17.63 ± 0.57
Uniform Score	$c_1 = 1.0$	8.54 ± 0.44	**10.10 ± 0.47**	2.01 ± 0.42	2.57 ± 0.40	6.67 ± 0.43	1.24 ± 0.29	3.11 ± 0.24	**10.00 ± 0.53**	5.03 ± 0.47	7.66 ± 0.28
	$c_2 = 0.8$	36.61 ± 0.33	61.73 ± 0.32	62.22 ± 0.33	**63.26 ± 0.39**	48.25 ± 0.24	43.12 ± 0.29	39.56 ± 0.37	55.85 ± 0.26	52.77 ± 0.24	49.81 ± 0.25

$\{0.5, 1.0\}$.[10] Due to the large strategy space, we focus on competitive meta-strategies that appear in the equilibria of their respective single-rescaling-factor meta-games. The base game uses symmetric costs of one, evaluated at $t = 10,000$.

Table 2 lists the meta-strategies and summarizes the main findings. The max-entropy MSNE consists of (RC, 0.5, f0.5) and (RC, 0.05, f0.5). We note that although (C, 0.5, f0.5) is best responded to by (C, 0.5, f1), yielding high collusion, (C, 0.5, f1) is more vulnerable to exploitation by RC strategies with $f = 0.5$. As a result, (C, 0.5, f1) achieves a high uniform score but also suffers high regret. Overall, CoI levels are lower when strategies with pessimistic initializations are included.

In effect, initialization can be interpreted as reflecting both an agent's adaptability and its prior beliefs about the opponent's behavior. These results highlight how the optimal choice of initialization and meta-strategy depends on beliefs about the opponent. If one expects its opponent to view cooperation as viable (e.g., optimistic initialization on both sides), adopting an optimistic initialization with a high learning rate is advantageous, leading to cooperative outcomes such as the PSNE (C, 0.5, f1). Conversely, if one anticipates an exploitative opponent, a more robust approach—pessimistic initialization and/or a low learning rate—is preferable to guard against exploitation.

We additionally construct a meta-game over C and RC strategies with a finer grid of learning rates to verify that the collusive outcome is robust to perturbations in α and does not rely on coordination of the hyperparameter choices. Further details can be found in Appendix C.6.

4.3 LLMs for Pricing

For LLMs, we adopt prompts similar to those in Fish et al. [23], augmented with additional strategy hints that serve as adaptation parameters. Each prompt consists of a constant and a variable component: the constant part specifies the adaptation strategy, while the variable part maintains a history of previous game states, the model's plan, and insight from the preceding round. This evolving variable part constitutes the internal representation Z, which is updated over time. The decoding and update functions, ϕ and ω, are determined jointly by the LLM and the constant portion of the prompt. Further details on prompt design are provided in Appendix E.2.

LLMs can be deployed at test time directly without pretraining, as in Fish et al. [23]. However, greater control over initial policies can be achieved by providing (simulated) interaction histories as part of the initial state context, placing this approach within the paradigm of in-context learning.

LLM pretraining. The goal of pretraining is to prepare an initial representation that encodes the initial policy to be used at test time. This representation includes historical interactions, the model's plan, and insight from the last pretraining round. Overall, the initial representation is jointly determined by the pretraining prompts, the paired opponent, and the historical interactions. During *pretraining*, the opponent's policy or characteristics are also described in the prompt to facilitate (pretraining) adaptation. The remaining prompt mirrors that used at test time. The amount of pretraining history included in the prompt is important, as LLM tends to place greater weight on recent rounds. Pretraining is terminated when the state remains unchanged for ten consecutive rounds, at which point the full history is included.

We consider three types of paired opponents during pretraining, giving four types of initial representations:

- *h0*: no history,
- *h1*: history of self-play against a symmetric LLM,
- *h2*: history of play against a pretrained Q-policy from the RC family as specified in Fig. 5, and
- *h3*: history of play against Tit-for-Tat, which also falls under the RC family in this setting.

LLM at test time. During inference, we keep the pretraining interaction history but remove the plan and insights, as they contain information specific to the pretraining opponent. An *adaptation strategy prompt* is then appended at test time to complete the inference prompt. We consider the following variants:

- *p0*: no strategy specified.
- *p1*: "Your co-participant may aim to learn an approximately best responding strategy to yours. Make sure your strategy achieves high profit even for the best responding strategies."

---

[10] $f = 0$ is excluded due to the very limited number of updates it induces.

Table 2: Meta-game evaluation of meta-strategies with different Q-value initializations, including $f = \{1, 0.5\}$ evaluated at $t = 10,000$.

$t = 10,000$	$f = 1$						$f = 0.5$					
Meta-strategies	C 0.5	C 0.05	C 0.005	RC 0.5	RC 0.05	RC 0.005	C 0.5	C 0.05	C 0.005	RC 0.5	RC 0.05	RC 0.005
PSNE	-	-	-	-	-	-	-	-	-	✓	✓	-
MSNE	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.37	0.00
NE-Regret ($\times 10^{-3}$)	10.20 ± 1.78	16.77 ± 1.31	16.88 ± 1.26	3.95 ± 1.67	8.84 ± 1.19	4.39 ± 1.37	4.03 ± 1.49	6.73 ± 1.53	13.58 ± 1.56	**0.00 ± 1.20**	**0.00 ± 1.41**	4.48 ± 1.40
Uniform Score	**32.41 ± 0.81**	21.14 ± 0.89	18.68 ± 0.90	31.90 ± 0.72	26.35 ± 0.62	26.76 ± 0.62	28.29 ± 0.87	24.63 ± 0.82	19.18 ± 0.84	31.47 ± 0.64	29.76 ± 0.66	27.04 ± 0.76

Table 3: Evaluation scores among LLM strategies evaluated at $t = 50$.

	p2h3	*p0h0*	*p1h2*	*p2h1*	*p1h1*	*p0h2*
PSNE	✓	-	-	-	-	✓
MSNE	0.44	0.00	0.00	0.00	0.00	0.56
NE-Regret ($\times 10^{-3}$)	**0.00 ± 7.92**	58.76 ± 7.40	47.38 ± 9.13	50.37 ± 9.28	59.47 ± 6.14	**0.00 ± 9.32**
Uniform Score	**36.79 ± 4.86**	10.84 ± 2.95	15.08 ± 3.83	14.54 ± 3.57	10.80 ± 2.57	25.55 ± 5.44

- *p2*: "One adaptation strategy is to try predicting the current strategy your co-participant uses and then update your strategy to approximately best respond to your co-participant."

p1 is designed to elicit robust policies with low update rates, whereas *p2* promotes adaptive behavior aimed at exploitation or cooperation. Broadly, the two meta-strategy dimensions can be interpreted as the choice of strategic prompt and the selection of historical information to include.

Recovering LLM's initial policy. The LLM's policy is shaped by both the constant and variable components of the prompt. For probing scalability, we consider the state as the price pair from one round, even though the prompt itself retains the full interaction history. To estimate the initial policy, we query the LLM 16 times per state using the same prompt. We use the resulting empirical price distributions as an approximation of the policy. The best response is then computed through value iteration [49].

Meta-strategies. We consider 24 meta-strategies in total, covering combinations of two LLM models (GPT5-mini and GPT5-nano), four types of interaction histories, and three distinct inference-time prompts. Fig. 23 in the appendix shows the locations of initial policies of these meta-strategies in the PC-CR space.

4.3.1 Meta-game Evaluation.

We focus on strategies generated by GPT5-mini, since GPT5-nano mostly converges immediately to the competitive price regardless of the paired opponent. Specifically, we select six meta-strategies that roughly span LC (*p0h0*, *p1h2* and *p2h1*), C (*p2h3*, *p1h1*), and RC (*p0h2*). Table 3 and Appendix Fig. 25 summarize main findings for the symmetric repeated pricing game with $c_1 = c_2 = 1$, evaluated at $t = 50$ with $N_{\text{discrete}} = 4$, $N_{\text{base}} = 40$.[11] Prices are discretized into four levels due to API costs, and initial states are uniformly sampled for the base games.

Interestingly, we find that *p2h3* and *p0h2* emerge as two pure-strategy Nash equilibria in the meta-game. The average payoffs of *p2h3* and *p0h2*, when playing against themselves and each other, are close to the payoffs under perfect collusion, suggesting that these strategies sustain cooperation, possibly due to pretraining histories that converge to collusion against policies in the RC family under Q-learning.

In contrast, *p0h0* plays the competitive price unless initialized in a collusive state.[12] Similar to the Q-learning comparison between (C, 0.5) and (RC, 0.005), although *p0h0* can exploit *p2h3*, such exploitation does not constitute a Nash equilibrium, since cooperation remains a profitable deviation. Taken together, these results suggest that collusion can emerge and persist as a stable outcome among rational LLM-based agents within the selected strategy set.

LLM-generated insights frequently emphasize keywords such as "cooperation", "trigger", and "punishment", and most strategies resemble Grim Trigger behavior, consistent with Fish et al. [23]'s observation that LLMs follow a steep reward–punishment scheme. As with Grim Trigger, interactions often settle into consistent competition once cooperation breaks down.

However, a novel pattern we observe is that certain strategies, most notably *p2h3*, can re-establish cooperation even after extended periods of competitive interaction. This recovery is opponent-dependent: when paired with *p0h2* or *p1h1*, *p2h3* gradually shifts play from competition back toward cooperation by persistently attempting collusive pricing. Unlike *p2h3*, *p0h2* quickly reverts to competitive pricing upon defection, e.g., when playing against *p1h1*, consequently underperforms relative to *p2h3* in these matchups. These observations suggest that, despite their recency bias, LLMs can draw on deeper historical context, even from pretraining, to strategically restore cooperation.

5 DISCUSSION

We developed a meta-game framework to evaluate whether collusion can emerge as a realistic outcome among strategically rational agents operating under test-time constraints. Each meta-strategy specifies an agent's choice of an initial policy family and in-game adaptation rule, allowing us to analyze strategic behavior beyond symmetric hyperparameter assumptions (a form of pre-game coordination).

Our results show that algorithmic collusion can persist in equilibrium among meta-strategies within Q-learning, UCB, and LLMs, suggesting that collusion may emerge from rational agents even in the absence of explicit communication. However, the extent of collusion depends on agents' beliefs about their opponents, e.g., pessimistic beliefs tend to yield low-collusion equilibria, whereas optimistic beliefs promote cooperative outcomes.

Natural extensions of our framework include modeling heterogeneous beliefs on the opponent's cost or representation initialization and analyzing through Bayesian Nash equilibrium, as well as broadening meta-strategy coverage, particularly for LLMs with richer prompt designs and pretraining histories. Additionally, cross-algorithm meta-games could shed further light on how algorithmic diversity affects the emergence and stability of collusion.

---

[11] We observe that all policies stop updating after about 30 rounds.

[12] Note that our results differ from Fish et al. [23] in that *p0h0* does not exhibit collusion for most initial states, even though the prompt design is similar. We were unable to replicate their findings due to model deprecations; we instead employ newer models (GPT5-mini, GPT5-nano, Gemini 2.5/2.0 flash/-lite).

ACKNOWLEDGMENTS

We thank the anonymous reviewers for constructive feedback. This work was supported in part by Rutgers SAS Research Grant in Academic Themes.

REFERENCES

[1] Ibrahim Abada, Joseph E. Harrington Jr, Xavier Lambin, and Janusz M Meylahn. 2024. Algorithmic Collusion: Where Are We and Where Should We Be Going? http://dx.doi.org/10.2139/ssrn.4891033

[2] Ibrahim Abada and Xavier Lambin. 2023. Artificial Intelligence: Can Seemingly Collusive Outcomes Be Avoided? *Management Science* 69, 9 (2023), 5042–5065.

[3] Ibrahim Abada, Xavier Lambin, and Nikolay Tchakarov. 2024. Collusion by mistake: Does algorithmic sophistication drive supra-competitive profits? *European Journal of Operational Research* 318, 3 (2024), 927–953.

[4] Marina Agranov and Leeat Yariv. 2015. Collusion through Communication in Auctions. *Games and Economic Behavior* 107 (2015), 93–108.

[5] Kushal Agrawal, Verona Teo, Juan J. Vazquez, Sudarsh Kunnavakkam, Vishak Srikanth, and Andy Liu. 2025. Evaluating LLM Agent Collusion in Double Auctions. arXiv:2507.01413 [cs.GT] https://arxiv.org/abs/2507.01413

[6] Eshwar Ram Arunachaleswaran, Natalie Collina, Sampath Kannan, Aaron Roth, and Juba Ziani. 2022. Algorithmic Collusion Without Threats. *arXiv:2409.03956* (2022).

[7] Stephanie Assad, Robert Clark, Daniel Ershov, and Lei Xu. 2024. Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market. *Journal of Political Economy* 132, 3 (2024), 723–771.

[8] David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. 2018. Re-evaluating evaluation. In *Advances in Neural Information Processing Systems*, Vol. 31.

[9] Martino Banchio and Andrzej Skrzypacz. 2022. Artificial Intelligence and Auction Design. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 30–31.

[10] Wolfram Barfuss and Janusz M Meylahn. 2023. Intrinsic fluctuations of reinforcement learning promote cooperation. *Scientific reports* 13(1) 1309 (2023).

[11] Martin Bichler, Julius Durmann, and Matthias Oberlechner. 2024. Online Optimization Algorithms in Repeated Price Competition: Equilibrium Learning and Algorithmic Collusion. arXiv:2412.15707 [cs.GT] https://arxiv.org/abs/2412.15707

[12] Gianluca Brero, Eric Mibuari, Nicolas Lepore, and David C. Parkes. 2022. Learning to Mitigate AI Collusion on Economic Platforms. In *Advances in Neural Information Processing Systems*, Vol. 35. 37892–37904.

[13] David P. Byrne and Nicolas de Roos. 2019. Learning to Coordinate: A Study in Retail Gasoline. *American Economic Review* 109, 2 (February 2019), 591–619.

[14] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. 2020. Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review* 110, 10 (October 2020), 3267–97.

[15] Cesare Carissimo, Fryderyk Falniowski, Siavash Rahimi, and Heinrich Nax. 2025. Algorithmic Collusion is Algorithm Orchestration. arXiv:2508.14766 [econ.TH] https://arxiv.org/abs/2508.14766

[16] Peter Cramton and Jesse A Schwartz. 2000. Collusive bidding: Lessons from the FCC spectrum auctions. *Journal of regulatory Economics* 17, 3 (2000), 229–252.

[17] Jacob W. Crandall. 2014. Towards Minimizing Disappointment in Repeated Games. *Journal of Artificial Intelligence Research* 49 (2014), 111–142.

[18] Jacob W. Crandall and Michael A. Goodrich. 2010. Learning to compete, coordinate, and cooperate in repeated games using reinforcement learning. *Machine Learning* 82 (2010), 281–314.

[19] Anthony DiGiovanni and Ambuj Tewari. 2022. Balancing Adaptability and Non-exploitability in Repeated Games. In *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*. 559–568.

[20] Nicolas Eschenbaum, Filip Mellgren, and Philipp Zahn. 2022. Robust Algorithmic Collusion. https://arxiv.org/abs/2201.00345

[21] Joshua Estelle, Yevgeniy Vorobeychik, Michael P. Wellman, Satinder Singh, Christopher Kiekintveld, and Vishal Soni. 2004. Strategic interactions in the TAC 2003 supply chain tournament. In *Proceedings of the 4th International Conference on Computers and Games*. 316–331.

[22] Ariel Ezrachi and Maurice E. Stucke. 2015. Algorithmic Collusion: Problems and Counter-Measures. (2015).

[23] Sara Fish, Yannai A. Gonczarowski, and Ran Shorrer. 2024. Algorithmic Collusion by Large Language Models. https://arxiv.org/abs/2404.00806

[24] James W Friedman. 1971. A non-cooperative equilibrium for supergames. *The Review of Economic Studies* 38, 1 (1971), 1–12.

[25] Aurélien Garivier and Eric Moulines. 2008. On Upper-Confidence Bound Policies for Non-Stationary Bandit Problems. arXiv:0805.3415 [math.ST] https://arxiv.org/abs/0805.3415

[26] Aurélien Garivier and Eric Moulines. 2011. On Upper-Confidence Bound Policies for Switching Bandit Problems. In *Algorithmic Learning Theory*. Springer Berlin Heidelberg, 174–188.

[27] Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčiak, The Anh Han, Edward Hughes, Vojtěch Kovařík, Jan Kulveit, Joel Z. Leibo, Caspar Oesterheld, Christian Schroeder de Witt, Nisarg Shah, Michael Wellman, Paolo Bova, Theodor Cimpeanu, Carson Ezell, Quentin Feuillade-Montixi, Matija Franklin, Esben Kran, Igor Krawczuk, Max Lamparth, Niklas Lauffer, Alexander Meinke, Sumeet Motwani, Anka Reuel, Vincent Conitzer, Michael Dennis, Iason Gabriel, Adam Gleave, Gillian Hadfield, Nika Haghtalab, Atoosa Kasirzadeh, Sébastien Krier, Kate Larson, Joel Lehman, David C. Parkes, Georgios Piliouras, and Iyad Rahwan. 2025. Multi-Agent Risks from Advanced AI. arXiv:2502.14143 [cs.MA] https://arxiv.org/abs/2502.14143

[28] Karsten T. Hansen, Kanishka Misra, and Mallesh M. Pai. 2021. Frontiers: Algorithmic Collusion: Supra-competitive Prices via Independent Algorithms. *Marketing Science* 40, 1 (2021), 1–12.

[29] Jason D. Hartline, Sheng Long, and Chenhao Zhang. 2024. Regulation of Algorithmic Collusion. In *Proceedings of the 2024 Symposium on Computer Science and Law*. 98–108.

[30] Jason D. Hartline, Chang Wang, and Chenhao Zhang. 2025. Regulation of Algorithmic Collusion, Refined: Testing Pessimistic Calibrated Regret. In *Proceedings of the 2025 Symposium on Computer Science and Law*. 108–120.

[31] Matthias Hettich. 2021. Algorithmic Collusion: Insights from Deep Learning. (2021). https://ssrn.com/abstract=3785966

[32] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. 2018. Is Q-Learning Provably Efficient?. In *Advances in Neural Information Processing Systems*, Vol. 31.

[33] Patrick R. Jordan, Christopher Kiekintveld, and Michael P. Wellman. 2007. Empirical game-theoretic analysis of the TAC Supply Chain game. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*.

[34] Christopher Kiekintveld and Michael P. Wellman. 2008. Selecting strategies using empirical game models: an experimental analysis of meta-strategies. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*. 1095–1101.

[35] Timo Klein. 2021. Autonomous algorithmic collusion: Q-learning under sequential pricing. *The RAND Journal of Economics* 52, 3 (2021), 538–558.

[36] Rohit Lamba and Sergey Zhuk. 2022. Pricing with algorithms. arXiv:2205.04661 [econ.TH] https://arxiv.org/abs/2205.04661

[37] Xavier Lambin. 2023. Less Than Meets the Eye: Simultaneous Experiments as a Source of Algorithmic Seeming Collusion. *SSRN Electronic Journal* (01 2023). https://doi.org/10.2139/ssrn.4498926

[38] Zun Li and Michael P. Wellman. 2024. A Meta-Game Evaluation Framework for Deep Multiagent Reinforcement Learning. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*. 148–156.

[39] Viliam Lisý and Michael Bowling. 2017. Eqilibrium Approximation Quality of Current No-Limit Poker Bots. In *The AAAI-17 Workshop on Computer Poker and Imperfect Information Games*.

[40] Carlos Martin and Tuomas Sandholm. 2024. ApproxED: Approximate exploitability descent via learned best responses. arXiv:2301.08830 [cs.GT] https://arxiv.org/abs/2301.08830

[41] Daniel McFadden. 1974. Conditional logit analysis of qualitative choice behavior. In *Fontiers in Econometrics*. Academic press, 105–142.

[42] J. M. Meylahn and L. Janssen. 2022. Limiting Dynamics for Q-Learning with Memory One in Symmetric Two-Player, Two-Action Games. *Complexity* 2022, 1 (2022), 483–491.

[43] Leon Musolff. 2022. Algorithmic Pricing Facilitates Tacit Collusion: Evidence from E-Commerce. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. ACM, 32–33.

[44] OECD. 2023. Algorithmic Competition, OECD Competition Policy Roundtable Background Note. (2023). www.oecd.org/daf/competition/algorithmic-competition-2023.pdf

[45] S. Phelps, M. Marcinkiewicz, and S. Parsons. 2006. A novel method for automatic strategy acquisition in N-player non-zero-sum games. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*. 705–712.

[46] Tabish Rashid, Bei Peng, Wendelin Boehmer, and Shimon Whiteson. 2020. Optimistic Exploration even with a Pessimistic Initialisation. In *International Conference on Learning Representations*.

[47] Daniel M. Reeves. 2005. Generating Trading Agent Strategies: Analytic and Empirical Methods for Infinite and Large Games. *PhD thesis, University of Michigan* (2005).

[48] Thomas C. Schelling. 1956. An Essay on Bargaining. *The American Economic Review* 46, 3 (1956), 281–306.

[49] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: An introduction*. MIT Press.

[50] Finbarr Timbers, Nolan Bard, Edward Lockhart, Marc Lanctot, Martin Schmid, Neil Burch, Julian Schrittwieser, Thomas Hubert, and Michael Bowling. 2022. Approximate Exploitability: Learning a Best Response. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. 3487–3493.

[51] Yuki Usui and Masahiko Ueda. 2021. Symmetric equilibrium of multi-agentreinforcement learning in repeated prisoner's dilemma. *Appl. Math. Comput.* 409 (2021), 126370.

[52] Yevgeniy Vorobeychik, Christopher Kiekintveld, and Michael P. Wellman. 2006. Empirical mechanism design: methods, with application to a supply-chain scenario. In *Proceedings of the 7th ACM Conference on Electronic Commerce.* 306–315.

[53] Elaine Wah and Michael P. Wellman. 2016. Latency arbitrage in fragmented markets: A strategic agent-based analysis. *Algorithmic Finance* 5 (2016), 69–93.

[54] Ludo Waltman and Uzay Kaymak. 2008. Q-learning agents in a Cournot oligopoly model. *Journal of Economic Dynamics and Control* 32, 10 (2008), 3275–3293.

[55] Xintong Wang, Christopher Hoang, and Michael P. Wellman. 2020. Learning-Based Trading Strategies in the Face of Market Manipulation. In *Proceedings of First ACM International Conference on AI in Finance.* 1–8.

[56] Xintong Wang, Yevgeniy Vorobeychik, and Michael P. Wellman. 2018. A Cloaking Mechanism to Mitigate Market Manipulation. In *Proceedings of 27th International Joint Conference on Artificial Intelligence.* 541–547.

[57] Xintong Wang and Michael P. Wellman. 2017. Spoofing the limit order book: An agent-based model. In *Proceedings of 16th International Conference on Autonomous Agents and Multiagent Systems.* 651–659.

[58] Michael P. Wellman. 2006. Methods for empirical game-theoretic analysis. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2 (AAAI'06).* AAAI Press, 1552–1555.

[59] Michael P. Wellman, Karl Tuyls, and Amy Greenwald. 2025. Empirical Game-Theoretic Analysis: A Survey. *Journal of Artificial Intelligence Research* 82 (2025). https://doi.org/10.1613/jair.1.16146