# A Critique of Reinforcement-Learning Approaches to Collusion in Oligopoly

Srikanth Pai* Arun Selvan Ramaswamy† Tania Mitra Victoria‡

December 2025

**Abstract**

A small number of firms competing to sell a product to a market constitutes an oligopoly. The firms control the market share by setting the price or controlling quantities. The game-theoretic study of this setting is well studied in economics. Collusion refers to firms sustaining prices above standard equilibrium benchmarks by responding to unilateral price cuts with actions that make deviation unprofitable. In recent years, with the advent of AI and the associated popularity of reinforcement learning (RL), pricing in oligopolies has been studied using these techniques. We integrate insights from both economic modeling and machine learning to clarify when and why algorithmic pricing yields supracompetitive outcomes. The paper reviews foundational economic models and the RL agent literature from recent times. We argue that the existence, convergence, and proper identification of collusion in settings with RL agents are far from settled.

## 1 Introduction and Motivation

Algorithmic pricing tools are increasingly deployed in digital markets, where sellers adjust prices frequently in response to competitors and demand conditions. Empirical studies of large e-commerce platforms, such as Amazon Marketplace, document that a substantial share of third-party sellers rely on automated pricing software rather than manual price-setting (Chen et al., 2016; Spann et al., 2025). In parallel, both economists and machine-learning researchers have begun to model these tools using reinforcement learning (RL), treating pricing as a sequential decision problem in which agents update prices based on realised profits and observed rival behaviour (Waltman and Kaymak, 2008; Calvano et al., 2020; Klein, 2021).

Recent years have witnessed remarkable progress in reinforcement learning (RL), achieving superhuman performance in complex sequential decision-making domains such as board games (Silver et al., 2016, 2017), real-time strategy environments (Vinyals et al., 2019; Ye et al., 2020), robotic control (Lillicrap et al., 2016; Haarnoja et al., 2018), and autonomous driving (Kendall et al., 2018; Codevilla et al., 2018). Much of this progress stems from the integration of deep neural networks as function approximators (Mnih

*Madras School of Economics, Chennai, India
†Karlstad University, Karlstad, Sweden
‡Madras School of Economics, Chennai, India

et al., 2015b; LeCun et al., 2015), enabling RL agents to operate in high-dimensional state and action spaces.

A significant fraction of these advances involve interaction among multiple adaptive agents, giving rise to the field of *multi-agent reinforcement learning* (MARL) (Busoniu et al., 2008; Hernandez-Leal et al., 2019; Zhang et al., 2021). In MARL, each agent optimizes its long-run reward in an environment influenced by the actions of others, often leading to emergent coordination, competition, or instability. Although this framework mirrors the strategic interactions long studied in economics, it is more popular in distributed control and robotics problems.

In economic theory, many markets are neither perfectly competitive nor fully monopolistic but instead consist of a few dominant firms whose actions are interdependent. Such markets are known as *oligopolies*. Each firm's optimal price or production decision depends on its expectations of how rivals will respond. Classical models formalize this dependence: the Cournot model treats firms as choosing quantities, the Bertrand model treats them as choosing prices, and the Edgeworth model introduces capacity constraints. These benchmark settings define the competitive landscape against which coordination or collusion can be understood.

Under full information and rational behaviour, the equilibria of these models are typically competitive: firms undercut each other until prices approach marginal cost. Sustained collusion—explicit or tacit—requires repeated interaction, credible punishment, and common knowledge of rationality (Fudenberg and Tirole, 1991; Tirole, 1988). These assumptions have guided antitrust reasoning for decades and underlie the belief that collusion cannot arise without communication or enforcement mechanisms.

Although interest in "algorithmic collusion" has grown rapidly, the term is often used loosely in the machine-learning literature to refer to any outcome with prices above the competitive benchmark. In economic theory, however, collusion has a far stricter meaning: firms must condition their behaviour on past actions and deter profitable deviations through credible punishment strategies (Stigler, 1964; Green and Porter, 1984; **?**). High prices alone are not sufficient. A central aim of this survey is to clarify this distinction and to evaluate existing RL results using the repeated-game notion of tacit collusion, which is the definition relevant for both economic theory and antitrust policy (Harrington, 2018). Making this distinction early is crucial, because many behaviours that look like collusion at the level of prices may instead arise from instability, optimisation failures, or artefacts of the learning algorithms themselves (Asker et al., 2022; Epivent and Lambin, 2024).

The existing RL literature on oligopoly pricing reflects this definitional ambiguity. In practice, learning agents display a wide range of behaviours: some converge to competitive prices, others oscillate, and some maintain elevated prices for long periods (Klein, 2021; Hettich, 2021; Dawid et al., 2024). These patterns must be interpreted with care, because high prices or cycles alone do not reveal whether the underlying policies implement the history-dependent incentives required for tacit collusion. For ML researchers, the key challenge is determining when observed outcomes reflect strategic interaction and when they arise from optimisation issues or artefacts of specific learning rules—such as asynchronous updating, temporal randomisation through replay buffers, or function approximation instability (Asker et al., 2022; Epivent and Lambin, 2024; **?**).

A recent survey by Abada et al. (2025) provides a comprehensive overview of algorithmic collusion, with emphasis on defining collusion outcomes, empirical relevance, and regulatory implications. Whereas their contribution is largely economic and policy-focused, our critique is more technical: we examine the specific reinforcement-learning al-

gorithms used in oligopoly pricing studies—tabular Q-learning, deep Q-networks, policy-gradient methods, and implementation details such as replay, synchronisation, and state design—and assess whether the behaviours reported in the literature satisfy the repeated-game notion of tacit collusion rather than arising from optimisation artifacts or instability.

We examine how different reinforcement-learning approaches behave in repeated pricing environments, how their design choices shape the dynamics they produce, and when the resulting behaviour should be interpreted as genuine strategic collusion in the economic sense. Equally important, we identify cases where elevated prices or apparent coordination arise from optimisation instability, exploration effects, or function-approximation artefacts rather than from history-dependent incentives. Our aim is to provide a clear framework for understanding what current RL models can and cannot tell us about tacit collusion in oligopolies, and to highlight the mechanisms that matter for future research at the intersection of economics and machine learning.

# 2 From Reinforcement Learning to Deep Reinforcement Learning

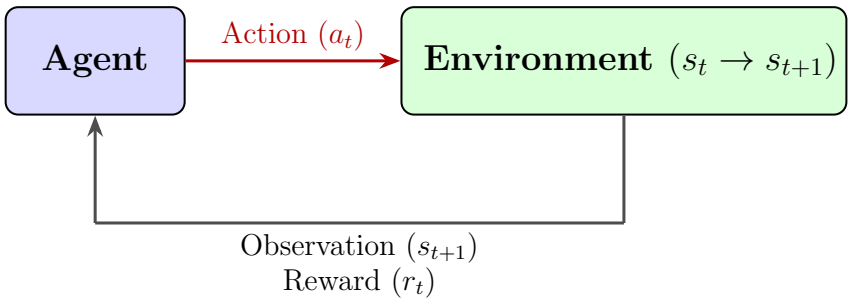


Figure 1: An agent interacts with an environment through actions. This internally causes the environment to change states. A local feedback is provided to the agent in terms of a reward and the next state. The agent picks the next action by taking the next state and the current reward into consideration.

Reinforcement Learning (RL) is a branch of Artificial Intelligence (AI) that deals with automating the sequential decision making task. It is closely linked to the fields of Stochastic Control and Dynamic Programming Bertsekas (2012); Bertsekas et al. (2011); Sutton and Barto (2018). The core idea is succinctly illustrated in Figure 1. An agent is tasked with finding a strategy to take decisions over time in lieu of solving a given sequential decision making problem. At any time $t$, the decision $a_t$ affects the environment in which the agents operate. The environment state $s_t$ changes to $s_{t+1}$. It then provides a feedback in the form of a reward $r_t$. The agent then refines its decision in the next step using $(r_t, s_{t+1})$. The goal of the agent, in RL, is to continuously interact with the environment and use the feedback to find a decision strategy that maximizes the future cumulative discounted rewards. In particular, find $\pi^*$ such that

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, \pi(s_t))\right], \tag{1}$$

where the expectation is taken with respect to the state transition kernel; $0 \leq \gamma \leq 1$ is the discount factor; suppose that we represent the set of environment states as $\mathcal{S}$ and we use $\mathcal{A}$ to represent the set of all possible actions then $\pi : \mathcal{S} \to \mathcal{A}$ represents the policy (decision map) that can be used by the agent to take a decision in a given (environment) state.

Typically in RL one is interested in time-homogeneous policies as in (1). In other words, one does not seek policies that vary with time. A policy $\pi$ can be associated with two key functions: (a) value function (b) Q-function. The value function associated with $\pi$ is given by $V^{\pi}(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, \pi(s_t)) \mid s_0 = s\right]$. Again, the expectation is taken with respect to the state transition kernel and the state state is fixed to be $s$. The optimal policy that maximizes the total discounted rewards is given by

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, V^{\pi}. \tag{2}$$

Similarly, the Q-function associated with $\pi$ is given by

$$Q^{\pi}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} V^{\pi}(s'), \tag{3}$$

where $s'$ is the next state given that action $a$ was taken in state $s$. The mathematical framework underlying RL is the Markov Decision Process (MDP). It is defined using the five tuple $\langle \mathcal{S}, \mathcal{A}, r, \mathcal{P}, \gamma \rangle$ , where $\mathcal{S}$ is the environment state space, $\mathcal{A}$ is the set of all possible actions, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S} \times \mathcal{A})$ such that $\mathcal{P}(\mathcal{S} \times \mathcal{A})$ is the space of all probability distributions over $\mathcal{S} \times \mathcal{A}$, and $0 \leq \gamma \leq 1$ is the discount factor.

Let us use $Q^*$ to represent the optimal Q-function $Q^{\pi^*}$, and $V^*$ to represent the optimal value function $V^{\pi^*}$. The following relation between these two quantities, called the Bellman equation is central to RL

$$Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} V^*(s'). \tag{4}$$

Further, $V^*(s') = \max_{a \in \mathcal{A}} Q(s', a)$ and $\pi^*(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q^*(s,a)$. Hence, the problem of finding an optimal policy reduces to finding the optimal Q-function.

** Before introducing deep Q-learning, it is helpful to recall the classical tabular Q-learning algorithm. In the single-agent setting, Q-learning estimates the optimal Q-function by iteratively updating its current approximation toward the Bellman target. After observing a transition $(s, a, r, s')$, the agent performs

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t \left[ r + \gamma \max_{a' \in \mathcal{A}} Q_t(s', a') - Q_t(s,a) \right], \tag{5}$$

where $\alpha_t$ is the learning rate. This stochastic approximation procedure converges to $Q^*$ under standard assumptions when the state–action space is finite (reference?). However, Q-learning becomes computationally infeasible in high-dimensional or continuous environments, motivating the use of function approximation through deep neural networks. **

Deep Q-Learning (DQL) Mnih et al. (2015a) is a very popular algorithm that uses a Deep Neural Network (DNN) called a Deep Q-Network (DQN) to approximate the optimal Q-function $Q^*$. Although simple, DQL is highly effective and does not succumb to Bellman's curse of dimensionality. One may argue that DQL is sufficient to solve a

multitude of discounted reward scenarios with discrete finite action spaces. The DQN is trained to minimize a loss derived from the Bellman equation (4) called the squared Bellman loss given by

$$\left(Q^*(s,a;\theta) - r(s,a) - \gamma \max_{a' \in \mathcal{A}} Q(s',a';\theta)\right)^2, \tag{6}$$

where $\theta$ represents the vector of DQN weights. The rest of the terms are as defined before. The goal, therefore, in DQL is to find $\theta^*$ that minimizes the above Bellman loss (6). This algorithm works very well for scenarios where the action space is discrete and finite. There are many problems, common in classical control, wherein the action space is continuous. Lillicrap et al. (2015) developed the Deep Deterministic Policy Gradient (DDPG) algorithm to address this issue. The algorithm was based on older results such as Maei et al. (2010) which explored the possibility of searching a parameterized policy space, guided by the objective to maximize the Q-function. In DDPG, an actor (policy neural network) iteratively refines its policy through constant feedback from a critic (DQN). Another policy algorithm that is based on searching the policy space is Proximal Policy Optimization (PPO), developed by Schulman et al. (2017). The primary difference between DDPG and PPO lies is the data used to train the policy network. While DDPG can learn from past experiences (data), PPO collects experiences using the current policy and then uses them to refine it. Algorithms like DDPG that can learn from past experiences are called as off-policy algorithms. On-policy algorithms are the ones that learn only from feedback obtained for current policy. The theory surrounding the asymptotic behavior of DQL has been studied in Ramaswamy and Hüllermeier (2021). There are many papers that have studied the finite time behavior of the above mentioned algorithms, such as Xu and Gu (2020); Qiu et al. (2021).

The hitherto discussed algorithms and framework focused on a single adaptive agent. In this view, any secondary agent is a part of the environment and therefore exhibits fixed behavior. This can be limiting in true multi-agent systems. The framework of Markov games expand the purview of MDPs to multi-agent scenarios Littman (1994). A Markov game is defined by a set of states $\mathcal{S}$, and a collection of $n$ action sets $\mathcal{A}_1, \ldots, \mathcal{A}_n$, one for each agent. The state transitions are controlled by a transition kernel $P : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_n \to \mathcal{P}(\mathcal{S})$. Every agent $i$ has an associated reward function $R_i : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_n \to \mathbb{R}$. This agent tries to maximize $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{i,t}\right]$, where $r_{i,t}$ is the single-stage reward obtained by agent $i$ at time $t$. The Multi-Agent Deep Deterministic Policy Gradient (MADDPG) developed by Lowe et al. (2017) is a popular multi-agent RL algorithm. The reader is referred to Yu et al. (2022); Redder et al. (2022); Zhang et al. (2021) for more details on multi-agent RL algorithms.

The remainder of the paper is organised as follows. Section 2 provides a concise overview of reinforcement learning methods relevant to pricing, with emphasis on the distinction between tabular and deep implementations and how these choices affect learning dynamics. Section 3 revisits standard oligopoly models (Bertrand, Cournot, and their dynamic extensions), which define the competitive and collusive benchmarks against which learning outcomes must be interpreted. Section 4 sets up repeated and dynamic interaction in oligopolies and explains how multi-agent reinforcement learning fits into this framework. Section 5 synthesises the RL-based pricing literature, evaluating when the reported behaviours satisfy the repeated-game notion of tacit collusion and when they are better understood as artefacts of particular algorithms or implementation choices. A

summary of published studies is presented in Table 5.

# 3 Classical Models of Oligopoly and Dynamic Pricing

Every firm ideally wants to be a monopoly, that is, sell their goods to the entire market. An oligopoly is a market with a small number of firms whose decisions are interdependent: each firm's optimal action depends on how its rivals are expected to respond. Firms can compete on prices, as in the Bertrand (1883) model, on quantities, as in Cournot (1838), or through product differentiation, as in Hotelling's (1929) framework. These canonical models are summarized briefly below. For more details, please look at Tirole (1988); Mas-Colell et al. (1995).

## 3.1 Single shot Cournot Model

In the Cournot setup, we assume there are $N$ firms competing to sell a single product and the consumers are indifferent to the companies producing the good. Each firm chooses the amount of quantity of goods to produce denoted by $q_i \geq 0$. The action profile is $\vec{q} = (q_1, \ldots, q_N)$, and total output is

$$Q = \sum_{j=1}^{N} q_j.$$

The market price is determined by an inverse demand function $P(Q)$, which is strictly decreasing. Firm $i$ incurs cost $C_i(q_i)$, which is usually assumed to be differentiable and convex. The profit (reward) to firm $i$ is

$$r_i(\vec{q}) = q_i\, P(Q) - C_i(q_i).$$

A Cournot–Nash equilibrium is the output profile in which each firm selects the quantity that maximizes its profit, taking the quantities of all other firms as given. From a reinforcement learning viewpoint, this is a single-state, continuous-action game in which each agent's reward depends on the aggregate choice $Q$.

## 3.2 Single shot Bertrand Model

In the Bertrand setup, firms choose prices instead of quantities. Each agent $i$ selects a price $p_i \geq 0$, forming the price profile $\vec{p} = (p_1, \ldots, p_N)$. Demand for firm $i$ is given by a function $D_i(p_i, p_{-i})$, which represents the quantities demanded by consumers across firms based on the price profile. With marginal cost $c_i$, the payoff to agent $i$ is

$$r_i(\vec{p}) = (p_i - c_i)\, D_i(p_i, p_{-i}).$$

A Bertrand–Nash equilibrium is a price vector in which each firm sets the price that maximizes its profit, taking the prices of all other firms as given. From a reinforcement-learning viewpoint, this is a single-state, continuous-action game where rewards depend on how prices jointly shape demand.

A widely used specification for the demand function $D_i$ is the *logit demand model* (Anderson et al., 1992):

$$D_i(\vec{p}) = \frac{\exp((a_i - p_i)/\mu)}{1 + \sum_{j=1}^{N} \exp((a_j - p_j)/\mu)},$$

where $\mu > 0$ is a constant.

# 4 Repeated and Dynamic Interaction for Oligopolies

We now discuss market evolution over time. Firms repeatedly observe competitors actions, update beliefs and adjust their strategies dynamically over time. Firms in dynamic pricing environments choose actions to maximise the present value of the firm

$$V = \mathbb{E}\Big[\sum_{k\geq 0} \gamma^k r^{(k)}\Big],$$

where $r^{(k)}$ is the payoff of the agent at time $k$. In the oligopoly literature, discount factor $\gamma$ represents the degree of 'non-myopia' of the firms. When $\gamma = 0$ the firms are said to be myopic. The expectation appears because future rewards depend on stochastic demands and, critically, on rivals' future actions, which appear random from the viewpoint of a single agent. Thus the dynamic oligopoly problem maps cleanly into a multi-agent RL setting where each firm learns a policy over a state summarising past prices and market conditions.

Multi-agent RL in oligopolies raises questions that do not appear in the one-agent setting. Independent Q-learning, using updates of the form (5), no longer has general convergence guarantees. Against the benchmark set by the one-shot Nash equilibrium and the monopoly outcome, three central questions arise:

1. Does the long-run pricing policy converge to the competitive Nash outcome ?
2. What is collusion and do the algorithms employed check for collusive behavior?
3. Is there a way to contain collusion in a digital marketplace by using learning algorithms?

In the remainder of the survey, we examine these questions using the current evidence from both the economics and machine-learning literature.

# 5 Emergence of Collusion in Oligopolies with Deep RL Agents

The notion of collusion is contested in both economics and competition law. Explicit collusion, understood as an agreement among firms that relies on inter-firm communication, is unambiguously illegal[1]. The difficulty lies in detecting tacit collusion, where firms coordinate purely through repeated actions without communication. Economics has long studied when repeated interaction allows firms to sustain tacit collusion. A Folk

[1]Sherman Act in the U.S., Competition Act in India

theorem by Fudenberg and Maskin (1986) shows that if the discount factor is sufficiently high and if firms can observe each other's behaviour, then many equilibrium outcomes become self-enforcing. In particular, firms can maintain high prices by rewarding cooperation and punishing deviations, even though no explicit agreement is made. These results imply that, in theory, repeated oligopoly environments naturally admit many stable, history-dependent strategies, so outcome-based definitions of collusion (e.g., "prices above Nash") are insufficient when analysing learning algorithms. Thus repeated-game theory characterizes when tacit collusion can arise as a self-enforcing equilibrium outcome. What theory does not tell us is whether multi-agent learning dynamics will actually find such strategies. In the context of machine-learning agents, the conditions under which specific algorithms discover collusive, history-dependent strategies remain unclear. Practically, with deep learning algorithms, there is also the possibility that supracompetitive prices are simply a consequence of optimisation failures arising from time, sample, or complexity restrictions.

Since supracompetitive prices (above one-shot Nash) are insufficient to define collusion, we need a policy-based perspective. Stigler (1964) argued that the central problem facing colluding firms is the deterrence of price cuts. In Stigler's spirit, we follow Harrington (2018) and Calvano et al. (2020) in considering collusion as "a reward–punishment scheme designed to provide the incentives for firms to consistently price above the competitive level." Thus collusion cannot be determined by outcomes alone but by the policies adopted by the firms. A necessary condition is that the strategy employed by a firm is history dependent. It is therefore of interest to understand whether machine-learning agents with their fixed, programmed learning rules are capable of producing such strategies.

One of the earliest papers that discusses collusion among multiple $Q$-learning agents in an oligopoly problem is Waltman and Kaymak (2008), who examined behaviour in a Cournot model. They varied the number of firms, the learning rate, the discount factor, and the memory of previously chosen quantities. Under linear demand and symmetric costs, average profits were significantly above Cournot–Nash but below monopoly levels across all settings.

Later, Calvano et al. (2020) were the first to investigate the reward–punishment definition of collusion in a Bertrand logit duopoly with tabular $Q$-learning agents. They used bounded memory and simultaneous pricing, and their forced-deviation experiments show a clear pattern: a price cut triggers a short punishment phase followed by a return to the supracompetitive price path. Their study was influential because it provided the first systematic evidence that simple learning dynamics can generate history-dependent pricing. At the same time, subsequent work raised questions about how to interpret these findings. Asker et al. (2022) show that asynchronous $Q$-learning can generate supracompetitive prices even in environments where no collusive mechanism is feasible, suggesting that some of the price elevation may be due to algorithmic structure rather than strategic behaviour. Epivent and Lambin (2024) further show that in the same environment, tabular $Q$-learning punishes not only downward deviations but also upward "invitations to collude," a pattern inconsistent with any rational collusive strategy and more consistent with learning instability as seen in Figure 2. These critiques concern interpretation rather than the empirical patterns themselves, but they highlight the need to test whether observed punishments are genuinely contingent on profitable deviations.

The authors of Calvano et al. (2020) used tabular $Q$-learning, a foundational yet simple algorithm. However, it suffers from the curse of dimensionality—the inability to handle large state spaces. Hettich (2021) extend Calvano et al.'s simultaneous-move


Figure 2: Reaction of tabular Q-learning agents following forced price deviations, taken from Epivent and Lambin (2024). In Calvano et al.'s setup, a downward deviation (cutting price) triggers a temporary retaliation phase, after which the high-price path is restored—consistent with reward–punishment behaviour as shown on the left. The graph on the right shows that *the same Q-learning agents also punish upward deviations*, i.e., when a firm unilaterally raises its price ("invitation to collude"). As Epivent and Lambin note, this is a surprising behaviour for colluding firms.

Bertrand duopoly to deep reinforcement learning by replacing tabular $Q$-learning with a Deep Q-Network (DQN). They also find that markets with more firms dilute collusion and that supracompetitive profits essentially disappear once the number of firms reaches 10; cf. Dawid et al. (2024). However, Hettich reports a short-lived price drop following a forced deviation and interprets this as evidence of a reward–punishment strategy. Unlike Calvano et al., he does not test whether this reaction is genuinely history dependent, whether deviations are unprofitable, or whether the behaviour persists over multiple periods or across different deviation magnitudes.

A related approach is taken by Han (2025), who modifies $Q$-learning to weight past experiences according to a firm's relative performance. This design can generate higher prices, but the analysis evaluates only final price paths and does not examine deviation responses or any form of punishment behaviour. Consequently, the study identifies supracompetitive outcomes but does not establish tacit collusion in the repeated-game sense.

Friedrich et al. (2024) extend Calvano-style Bertrand–Logit pricing to an episodic, inventory-constrained Markov game and report supra-competitive outcomes for DQN and PPO agents using a profit-based collusion index; their forced deviation tests, however, reveal only mild and almost profit-neutral reactions, so the evidence for robust, history-dependent punishment is considerably weaker than in Calvano et al. and closer to outcome-based collusion measures than to repeated-game notions of tacit collusion

A natural extension of these environments introduces uncertainty in the observation of rivals' actions. In repeated-game theory, imperfect monitoring settings are known to generate endogenous price wars, as in the classic model of Green and Porter (1984). An RL-based counterpart of this mechanism is addressed by Calvano et al. (2021), who adapt Q-learning agents to a Green–Porter-style environment in which demand shocks confound price deviations. They show that when learning time is sufficiently long, Q-learning agents internalize the monitoring friction and still achieve supracompetitive outcomes sustained by finite-duration punishment cycles. In contrast with perfect-monitoring setups, pun-

ishments do not only follow deliberate deviations but also occur after adverse shocks.

Unlike Calvano et al.'s simultaneous-move setup, Klein (2021) study a setting where firms update prices one at a time (a sequential repeated game). They show that tabular $Q$-learning can produce two kinds of behaviour depending on the coarseness of the price grid. With a small, discrete set of allowable prices, the algorithms often settle on a stable supracompetitive price and exhibit clear reward–punishment strategies. When the price grid is finer, the learning process instead produces repeated downward drift followed by sharp upward resets, and these cycles keep the average price above the competitive benchmark.

A separate line of work evaluates how different learning rules and implementation choices determine long-run pricing outcomes. Deng et al. (2024) compare DQN, PPO, SAC and related deep RL methods and find that these architectures usually converge either to competitive prices or to unstable cycles, and none display credible deviation–punishment responses. Studies that report elevated prices without testing deviation responses, such as Schlechtinger (2024), cannot establish tacit collusion since they document high or oscillatory prices but do not determine whether a unilateral deviation would be punished. Taken together, the evidence indicates that while tabular Q-learning can generate genuine history-dependent punishments in restricted environments, the behaviour of deep RL agents is typically explained by instability or optimisation artifacts rather than tacit collusion.

Since it looks plausible that learning agents collude, a natural line of questioning is whether the rules of the environment can be modified to counteract collusion. Johnson et al. (2023) show that in digital marketplaces, steering consumer visibility towards firms that deviate downward can eliminate collusive practices among Q-learning agents, and that when such steering depends on past pricing histories, supracompetitive equilibria collapses. Brero et al. (2022) treat the platform as a reinforcement learning agent who is a Stackelberg leader and learn rules that choose which seller to display based on price history. In this work, the buyers are Q-learning agents. They demonstrate that this method is effective in containing collusion compared to steering rules designed by hand.

A summary of peer reviewed papers that discuss RL agents used by firms to set prices is presented in Table 5.

| Paper | Model | Move | Algo | State repr. | FD? | Punish? | Main finding |
|---|---|---|---|---|---|---|---|
| Waltman and Kaymak (2008) | C | Sim | TQL | None | No | No | Profits > Nash; no strategic response |
| Calvano et al. (2020) | BL | Sim | TQL | Past price pairs | Yes | Yes | Clear reward–punishment equilibrium path |
| Klein (2021) | B | Seq | TQL | Price grid + memory | Yes | Yes* | Collusion only on coarse grid; cycles otherwise |
| Han (2025) | B | Sim | WQL | Weighted replay | No | No | Higher prices; deviation responses not tested |
| Asker et al. (2022) | BLv | Async | TQL | None | Yes | Weak | Supracompetitive even when collusion infeasible |
| Epivent and Lambin (2024) | BL | Sim | TQL | Bounded memory | Yes | Yes† | Punishes both upward and downward deviations |
| Johnson et al. (2023) | BL | Sim | TQL | History window on rivals | Yes | Indirect | Platform steering collapses supracompetitive outcomes |
| Schlechtinger (2024) | B | Sim | DQN/PPO | Own price only | No | No | Supra-competitive prices even without observing rivals; oscillatory pricing |
| Friedrich et al. (2024) | BL + inv | Sim | DQN/PPO | Last prices + inventories | Yes | Weak | Episodic, inventory-constrained market; DQN/PPO reach supra-competitive prices with mild, short-lived reactions to forced deviations |

Table 1: Summary of peer-reviewed RL-agent based seller pricing studies.

**Model codes:** BL = Bertrand–Logit, B = Bertrand, C = Cournot, BLv = variant of Bertrand–Logit.
**Move:** Sim = simultaneous price updates, Seq = sequential updates, Async = asynchronous updates.
**Algorithms:** TQL = tabular Q-learning, WQL = weighted Q-learning, DQN = Deep Q-network.
**FD?** indicates whether a forced-deviation experiment was performed.
**Punish?** indicates whether deviations were followed by a return to the supracompetitive path.
* Punishment verified only for coarse price grids. † Upward deviations also punished, which contradicts rational collusion.

# 6 Conclusion

Current evidence shows that reinforcement-learning agents can generate price trajectories above static Nash outcomes in small oligopolies. However, these patterns rarely meet the repeated-game criterion for tacit collusion. Only bounded-memory tabular Q-learning, under simultaneous moves and explicit forced-deviation tests, exhibits credible reward–punishment responses, and even those are fragile. In most deep-RL implementations, elevated prices arise without demonstrable deviation deterrence; when deviations are induced, responses are weak, symmetric with respect to upward and downward moves, or collapse quickly.

Our assessment is that the literature substantially over-interprets outcome-level price elevation as collusion. Collusion claims must instead rest on history-dependent continuation behaviour where deviations reduce discounted value relative to compliance. Very few studies satisfy this test. Future work should benchmark algorithms against explicit profitability-based deviation checks, rather than price paths alone, before making claims of autonomous collusion.

# References


Abada, I., Harrington, J. E., Lambin, X., and Meylahn, J. M. (2025). Algorithmic collusion: Where are we and where should we be going? Technical report, SSRN Working Paper. Available at SSRN: https://ssrn.com/abstract=4891033.

Anderson, S. P., de Palma, A., and Thisse, J.-F. (1992). *Discrete Choice Theory of Product Differentiation*. MIT Press, Cambridge, MA.

Asker, J., Fershtman, C., and Pakes, A. (2022). Artificial intelligence, algorithm design, and pricing. *AEA Papers and Proceedings*, 112:452–456.

Bertsekas, D. (2012). *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific.

Bertsekas, D. P. et al. (2011). Dynamic programming and optimal control 3rd edition, volume ii. *Belmont, MA: Athena Scientific*, 1.

Brero, G., Mibuari, E., Lepore, N., and Parkes, D. C. (2022). Learning to mitigate AI collusion on economic platforms. In *Advances in Neural Information Processing Systems*, volume 35, pages 37892–37905.

Busoniu, L., Babuska, R., and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(2):156–172.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–3297.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2021). Algorithmic collusion with imperfect monitoring. *International Journal of Industrial Organization*, 79:102712.

Chen, L., Mislove, A., and Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th International Conference on World Wide Web*, page 1339–1349. International World Wide Web Conferences Steering Committee.

Codevilla, F., Müller, M., López, A., Koltun, V., and Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dawid, H., Fichtner, A., and Harting, P. (2024). Deep reinforcement learning in oligopolistic markets: Stability and collusion. *Journal of Economic Dynamics and Control*. Forthcoming.

Deng, S., Schiffer, M., and Bichler, M. (2024). Algorithmic collusion in dynamic pricing with deep reinforcement learning. *arXiv preprint arXiv:2406.02437*.

Epivent, A. and Lambin, X. (2024). On algorithmic collusion and reward–punishment schemes. *Economics Letters*, 237:111661.

Friedrich, P., Pásztor, B., and Ramponi, G. (2024). Collusion of reinforcement learning-based pricing algorithms in episodic markets. In *Proceedings of the Agentic Markets Workshop at ICML 2024*. Poster.

Fudenberg, D. and Maskin, E. (1986). The folk theorem in repeated games with discounting or with incomplete information. *Econometrica*, 54(3):533–554.

Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT Press.

Green, E. J. and Porter, R. H. (1984). Noncooperative collusion under imperfect price information. *Econometrica*, 52(1):87–100.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

Han, B. (2025). Algorithmic pricing with independent learners and relative experience replay. In *Proceedings of the ACM International Conference on AI in Finance (ICAIF)*. Forthcoming.

Harrington, J. E. (2018). Developing competition law for collusion by autonomous artificial agents. *Journal of Competition Law & Economics*, 14(3):331–363.

Hernandez-Leal, P., Kartal, B., and Taylor, M. E. (2019). A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797.

Hettich, M. (2021). Algorithmic collusion: Insights from deep learning. Technical Report 94/2021, Center for Quantitative Economics (CQE), University of Münster.

Johnson, J. P., Rhodes, A., and Wildenbeest, M. (2023). Platform design when sellers use pricing algorithms. *Econometrica*, 91(5):1841–1879.

Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J., Lam, V. D., Bewley, A., and Shah, A. (2018). Learning to drive in a day. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

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

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–444.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., and Silver, D. (2016). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

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

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Maei, H. R., Szepesvári, C., Bhatnagar, S., and Sutton, R. S. (2010). Toward off-policy learning control with function approximation. In *ICML*, volume 10, pages 719–726.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press, New York.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015a). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., and Hassabis, D. (2015b). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Qiu, S., Yang, Z., Ye, J., and Wang, Z. (2021). On finite-time convergence of actor-critic algorithm. *IEEE Journal on Selected Areas in Information Theory*, 2(2):652–664.

Ramaswamy, A. and Hüllermeier, E. (2021). Deep q-learning: Theoretical insights from an asymptotic analysis. *IEEE Transactions on Artificial Intelligence*, 3(2):139–151.

Redder, A., Ramaswamy, A., and Karl, H. (2022). Multi-agent policy gradient algorithms for cyber-physical systems with lossy communication. In *ICAART (1)*, pages 282–289.

Schlechtinger, M. (2024). Quantifying collusion in a market simulation with deep reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., and Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550:354–359.

Spann, M., Bertini, M., Koenigsberg, O., Zeithammer, R., Aparicio, D., Chen, Y., Fantini, F., Jin, G. Z., Morwitz, V. G., Popkowski Leszczyc, P., Vitorino, M. A., Williams, G. Y., and Yoo, H. (2025). Algorithmic pricing: Implications for marketing strategy and regulation. *International Journal of Research in Marketing*.

Stigler, G. J. (1964). A theory of oligopoly. *Journal of Political Economy*, 72(1):44–61.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, second edition.

Tirole, J. (1988). *The Theory of Industrial Organization*. MIT Press.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., and Silver, D. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575:350–354.

Waltman, L. and Kaymak, U. (2008). Q-learning agents in a cournot oligopoly model. *Journal of Economic Dynamics and Control*, 32(10):3275–3293.

Xu, P. and Gu, Q. (2020). A finite-time analysis of q-learning with neural network function approximation. In *International conference on machine learning*, pages 10555–10565. PMLR.

Ye, D. et al. (2020). Towards playing full moba games with deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624.

Zhang, K., Yang, Z., and Basar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*, pages 321–384. Springer.