Algorithmic Collusion in Dynamic Pricing: Past, Present, and Future*

Chen Tang

July 2024

School of Data Science, Chinese University of Hong Kong, Shenzhen
ChenTang@link.cuhk.edu.cn
ChenTang01.github.io

This survey examines the development of research on algorithmic collusion in competitive dynamic pricing. Past numerical research has indicated that AI algorithms can lead to collusion. Some empirical and analytical evidence also supports this finding. Although the evidence is incomplete, the apparent algorithmic collusion could potentially threaten social welfare. However, most studied cases of algorithmic collusion are either *spurious* or *genuine* collusion, indicating that such collusion can be broken despite the presence of a reward-punishment scheme in some settings. A more robust form of algorithmic collusion, termed *authentic* collusion, has been rigorously defined, and several authentic collusive algorithms have been developed. Furthermore, I review the regulatory measures against algorithmic collusion and suggest future directions in this field.

Key words: Reinforcement Learning, Tacit Collusion, Spurious Collusion, Genuine Collusion, Authentic Collusion, Market Design, Operations Management

1. Introduction

Dynamic pricing studied how to adapt prices in response to demand fluctuations and is usually employed in settings where prices can be modified frequently. According to a report by McKinsey,

* I'm grateful to Prof. Guillermo Gallego from the Chinese University of Hong Kong, Shenzhen for his encouragement during my writing. I'm also grateful to Prof. Zhijie Tao from the Shanghai University of Finance and Economics. My research experience with him inspired me to write this paper.

I write a tutorial which reproduce part of the results of Calvano, Calzolari, Denicolo, and Pastorello (2020), Hansen, Misra, and Pai (2021).

Any feedback, suggestions, and corrections from readers are sincerely welcome.

implementing dynamic pricing solutions in most business practices has resulted in a sales growth of 2 to 5 percent and an increase in profit margins of 5 to 10 percent (BenMark, Klapdor, Kullmann, and Sundararajan (2017)). Given the substantial economic advantages offered by dynamic pricing strategies, dynamic pricing solutions have found widespread application across various industries. Notably, sectors such as the electricity market (Dutta and Mitra (2017)), online retailing (Lei, Jasin, and Sinha (2018)), and airline companies (Wittman and Belobaba (2019)) have embraced these solutions to improve revenues.

It's worth noting that dynamic pricing, contrary to a common intuition that it merely squeezes the surplus from consumers, has been shown to improve consumer welfare and result in a win-win outcome (N. Chen and Gallego (2019),Williams (2022)). In practice, it can lift consumer satisfaction (BenMark et al. (2017)).

Dynamic pricing has received significant interest from scholars of various disciplines, including industrial organization, game theory, management science, quantitative marketing, and computer science. Past research on dynamic pricing has primarily concentrated on pricing strategies, demand learning, and pricing considering various scenarios such as inventory constraints and strategic consumer behaviors. den Boer (2015) provides a comprehensive overview of dynamic pricing. Additionally, Gallego and Topaloglu (2019) is recommended as a textbook for conducting research in dynamic pricing, and Özer and Phillips (2012) serves as a valuable handbook for understanding dynamic pricing practices.

Most prior research has commonly assumed the monopoly situation, which does not align with real-world conditions. Despite ongoing debate about whether incorporating competitors can enhance the performance of pricing models, numerous studies have explored this issue (see Subsection 1.1 for more details). It is broadly recognized that competition tends to drive down prices and increase consumer welfare.

Using algorithms to price enables firms to respond more quickly to competitors' decisions in competitive markets (Brown and MacKay (2023)). However, a question arises when all firms employ AI algorithms, such as reinforcement learning, to set prices: could this lead to a monopoly market outcome? The answer may indeed be positive, and if the algorithms lead to a monopoly price, understanding the causes and exploring how to prevent diminishing social welfare is critical. These questions have given rise to a burgeoning interdisciplinary field known as **algorithmic collusion**, which has garnered significant attention since 2020.

The recent study of algorithmic collusion originated with the work of Calvano, Calzolari, Denicolo, and Pastorello (2020)[1], in which the authors demonstrated through numerical experiments that the use of Q-learning can result in supra-competitive pricing outcomes within a competitive market setting. The term 'supra-competitive' refers to prices higher than the one-stage competitive equilibrium. In a society where an increasing number of companies are now developing AI algorithms for real-time dynamic pricing (Spann et al. (2024), BCG (2020)), the potential of algorithmic collusion poses a huge threat to social welfare.

To provide readers with an initial taste of algorithmic collusion, I will employ the classic toy model—the prisoner's dilemma—as an illustrative example. Despite its simplicity, this model effectively captures the essence of the collusion scenario. Consider a static game involving two firms, indexed as 1 and 2, selling a homogeneous product. Each firm has the option to set its price at one of two levels: high (H) or low (L). The payoff matrix is given in table 1: In this single-stage

Table 1 The payoff matrix of the toy model

		Player 2	
		H	L
Player 1	H	$(2, 2)$	$(0.5, 3)$
	L	$(3, 0.5)$	$(1, 1)$

game, the sole Nash equilibrium is for both firms to set low prices. This outcome exemplifies the benefits of competition, as it leads to cheaper prices for consumers. However, Calvano, Calzolari, Denicolo, and Pastorello (2020) demonstrated that when the payoff matrix is unknown to the firms, and both firms utilize AI algorithms to adjust their prices dynamically, they will eventually learn to collude, ultimately setting high prices. This phenomenon is called algorithmic collusion[2].

Now, let's consider the same game played infinitely over time, with the two firms having discount rates of δ_1 and δ_2, respectively, where $0 < \delta_1, \delta_2 < 1$. Consider the one-shot deviation strategy (Hendon, Jacobsen, and Sloth (1996)), i.e. both firms initially cooperate by price H, but if one observes the other playing L, she will retaliate by also playing L for the remainder of the game. This strategy can serve as a credible threat to deter deviation, as long as the expected payoff from not deviating is greater than the payoff from deviation. In this toy model, a Subgame Perfect

[1] This paper has many versions. It was initially published in 2019 (Calvano, Calzolari, Denicolò, and Pastorello (2019)). A more detailed version was released in 2020 (Calvano, Calzolari, Denicolo, and Pastorello (2020)) and garnered significant attention from the academic community. Subsequently, a condensed version was published (Calvano, Calzolari, Denicolò, Harrington Jr, and Pastorello (2020)).

[2] This explanation is intended to provide a basic conceptual understanding for readers less familiar with algorithmic collusion. In subsection 3.1, a more detailed and comprehensive definition of algorithmic collusion will be presented.

Nash Equilibrium where both firms price H exists when $\delta_1, \delta_2 \geq \frac{2}{3}$.[3] The game can lead to a collusive outcome when the discount factors are sufficiently high. This aligns with the Folk theorem (Friedman (1971)), which states that collusion can occur as long as players are patient enough.

I refer to the one-shot deviation principle and the Folk theorem to illustrate that for any rational collusion to be sustained, one firm must be able to deter the other from deviating by lowering prices. In this example, a 'smart' firm 1 needs to possess a credible threat to punish firm 2 for deviating from the collusive strategy (H, H) to the non-cooperative strategy (H, L), and vice versa. While some AI algorithms do result in supra-competitive outcomes, this may simply be because they are not aware that lowering prices to (H, H) would yield the highest immediate profit (bearing in mind that in these studies, the payoff matrix is always not observable to the firms). Calvano, Calzolari, Denicoló, and Pastorello (2023) categorizes algorithmic collusion into two types:

- *Spurious Collusion*: Here, high prices are the result of firms not recognizing the potential for greater short-term profits through deviation, rather than due to an explicit reward-punishment scheme.
- *Genuine Collusion*: In this case, high prices are maintained through a reward-punishment mechanism where the AI algorithm exhibits behavior akin to the one-shot deviation principle.

In fact, the definitions of genuine and spurious collusion presented here are not exhaustive, and there is ongoing debate among scholars about whether so-called 'genuine' collusion is truly authentic. For instance, numerous numerical experiments discussed in section 2 demonstrate that players can threaten punishment for deviation, but they may not be sufficiently 'smart' to act as absolutely rational agents. The seeming one-shot deviation principle some AI pricing algorithms exhibit may have vulnerabilities that can be exploited, thereby precluding them from being classified as collusion.

After observing the weakness of genuine collusion, some research has refined the concept of algorithmic collusion (den Boer (2023)). In this paper, I refer to this refined concept as authentic collusion. [4]

- *Authentic Collusion*: This refers to collusion sustained by a theoretically reliable threat, and the collusive algorithm performs well enough against alternative algorithms.

[3] $2 \cdot \sum_{t=0}^{\infty} \delta_1^t \geq 3 + 1 \cdot \sum_{t=1}^{\infty} \delta_1^t, \quad 2 \cdot \sum_{t=0}^{\infty} \delta_2^t \geq 3 + 1 \cdot \sum_{t=1}^{\infty} \delta_2^t$

[4] The term 'authentic collusion' is named by myself, and this definition is simplified from den Boer (2023). A detailed illustration of this concept will be provided in Section 3.

In Subsection 3.1, an introduction is presented to the definitions of these three types of collusion, set within a more structured framework. Many past numerical studies have shown that AI algorithms can converge to spurious or genuine collusion in pricing games. This statement is also supported by empirical and analytical findings. Authentic collusive algorithm has also been developed (Meylahn and V. den Boer (2022), Loots and den Boer (2023)).

It is crucial to emphasize that there is often a mischaracterization regarding the distinction between authentic collusion and genuine collusion. This misinterpretation is one of the key reasons for the ongoing debate on whether algorithmic collusion poses a significant threat to market competition. It is also important to recognize that genuine collusion, while perhaps not embodying a perfectly rational colluder, does not imply that it should be regarded as harmless to market fairness. Keep in mind that the distinctions between spurious collusion, genuine collusion, and authentic collusion are **not** merely tautology but a meaningful issue.

To the best of my knowledge, this is the first paper to provide a comprehensive review of the evolution and development of algorithmic collusion. Related works include Assad et al. (2021) and Dorner (2021), but this work integrates research from broader disciplines and provides a more comprehensive and systematic examination of algorithmic collusion, along with potential opportunities for future research. The structure of this paper is organized as follows: Section 2 delves into the development and debates surrounding algorithmic collusion under reinforcement learning algorithms. Section 3 explores the research on designing algorithms that can achieve authentic collusion. Section 4 focuses on regulatory approaches against algorithmic collusion. Finally, in section 5, I will summarize past progress and suggest directions for future research.

1.1. Dynamic Pricing under Competition

Here I briefly review competitive dynamic pricing; for a more detailed illustration, see M. Chen and Chen (2015). Much of the prior research in this area has centered on the theoretical equilibria of pricing games under various settings such as inventory capacity constraints, strategic consumer behaviors, and product differentiation. For example, Adida and Perakis (2010) offers solutions for calculating the normalized Nash equilibrium in a dynamic scenario where multiple products share production capacity. Martínez-de-Albéniz and Talluri (2011) investigates the equilibrium of pricing games where each seller faces a fixed inventory capacity. Gallego and Hu (2014) examines

an oligopoly market dealing with a range of perishable, differentiated products and discovers that, under deterministic demand, the equilibrium can exhibit a simple structure: the equilibrium prices at any given time can be determined by a one-shot price competition game based on the current demand, while also considering a set of time-invariant shadow prices that reflect the aggregate capacity externalities.

Another line of research concentrates on the validity of the *market response hypothesis* in competitive markets. This hypothesis conjects that even if a firm models the market as a monopoly and learns the model parameters from historical data, the monopoly model can inherently capture information about competitors through market dynamics. This stream is related to misspecified learning (Besbes and Zeevi (2015)). For example, Cooper, Homem-de-Mello, and Kleywegt (2015) studies that, in a duopoly market, if both firms use a monopoly model to guide the pricing, how would the game converge? They found that under some settings, the result of applying the monopoly model would not converge to the static Nash equilibrium. Finally, some additional papers delve into pricing strategies as a response to competitors, such as Fisher, Gallino, and Li (2018) and Schlosser and Boissier (2018).

In summary, the study of algorithmic collusion of competitive dynamic pricing is a relatively new field, with work in this area gaining momentum after 2019.

2. Algorithmic Collusion under Reinforcement Learning

In this section, the fundamentals of reinforcement learning algorithms are outlined, with a focus on Q-learning. Most past research in this area is numerical experiments and will be detailed in Subsection 2.2, along with criticize in 2.3. Then 2.4 and 2.5 will provide theoretical and empirical evidence for algorithmic collusion respectively. Finally, I will give my remarks on the algorithmic collusion under RL in Subsection 2.6,.

2.1. Introduction to Reinforcement Learning

Reinforcement learning has its roots in the Markov decision process (MDP), which serves as a mathematical framework for modeling discrete-time sequential decision-making processes. Within this framework, a decision-making process can be characterized by five key elements:

1. States ($\mathcal{S}$): a finite set representing all possible states of the environment;
2. Actions ($\mathcal{A}(\mathcal{S})$): a finite set of actions available to the decision-maker, which may depend on the current state;
3. Transition Probabilities ($\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$): The probability of transitioning to a new state given the current state and the action taken;
4. Rewards Function ($\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$): A function that assigns a reward for each state-action pair;
5. Discount Factor ($\delta \in [0, 1]$): A factor that discounts future rewards based on their distance from the present.

During each discrete time t in a state $s_t \in \mathcal{S}$, the decision maker selects an action $a_t \in \mathcal{A}(s_t)$ and receives a reward r_t according to the rewards function $\mathcal{R}$. The agent then transitions to a new state s_{t+1} based on the transition probabilities $\mathcal{T}$. A policy π is defined as a mapping from states to actions, and the goal of an agent is to find the optimal policy that maximizes the expected cumulative reward over time, conditioning on the environment's transition probabilities $\mathcal{T}$ and rewards function $\mathcal{R}$, as well as the current state s_t:

$$\max_{\pi} \mathbf{E}\left[\sum_{i=t}^{\infty} \delta^i r_i \mid \pi, \mathcal{T}, \mathcal{R}, s_t\right]$$

Reinforcement learning is designed to find the optimal policy within an MDP process through the interactions between an agent and its environment. The majority of reinforcement learning algorithms operate on a *model-free basis*, signifying that they do not require prior knowledge of the environment's dynamics, such as the transition probabilities and the reward function. These algorithms also do not rely on maintaining an estimation of the environment to make decisions. One of the most prominent methods in RL is Q-learning, which maintains a Q-function $Q(s, a)$ for every state-action pair. In the context of Q-learning, the optimal policy is always to pick the action associated with the highest Q-value:

$$\pi(s) = \arg\max_{a} Q(s, a)$$

The Q-function represents the expected return obtained after taking action a in state s and following the optimal policy thereafter. The core idea of Q-learning is to iteratively update the Q-function using the Bellman equation, which relates the value of a state-action pair to the immediate reward and the discounted value of the next state:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t(s_t, a_t) + \delta \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t)], \tag{2.1}$$

where α is the **learning rate** adjusting the weight of new information. When a policy is chosen by maximizing the Q-function, this policy is deemed optimal, and thus the current Q-function can be referred to as the optimal Q-function. It has been proven that the Q-function will converge to the optimal Q-function if every state-action pair is visited an infinite number of times (Melo (2001)).

It should be noted that Q-learning can only achieve convergence to the optimal policy under the assumption that every state-action pair is visited infinitely often. This leads to a key aspect of Q-learning: balancing exploration and exploitation. Choosing the action with the highest Q-value is an act of greediness, which is termed as *exploitation*. In the Q-learning algorithm, there is a constant concern regarding whether the estimated Q-function is indeed optimal, necessitating *exploration*. To address this, Q-learning employs a strategy known as ϵ-greedy. At each time step t, the agent selects an action randomly with a probability of ϵ, and with a probability of $1-\epsilon$, it chooses the action associated with the highest Q-value. The algorithm process of Q-Learning with ϵ-greedy exploration is illustrated in Algorithm 1.

Algorithm 1 Q-Learning with ϵ-Greedy Exploration

Initialize action-value function $Q(s,a)$ arbitrarily for all s in $\mathcal{S}$ and a in $\mathcal{A}$
Initialize $Q'(s,a) \leftarrow 0$
Initialize a learning rate α, discount factor δ, exploration rate ϵ, and threshold δ
$s \leftarrow$ initial state
while $|Q - Q'|_\infty > \delta$ **do**
    $Q \leftarrow Q'$
    $a \leftarrow$ **argmax** $Q(s,a)$ with probability $1-\epsilon$ ▷ Exploitation
    $a \leftarrow$ random action from $\mathcal{A}$ with probability ϵ ▷ Exploration
    Take action a, observe reward r and next state s'
    $Q'(s,a) \leftarrow Q(s,a) + \alpha(r + \delta \cdot \max_{a'} Q(s',a') - Q(s,a))$
    $s \leftarrow s'$ ▷ Transition to the next state
end while
    return $\pi^*(s) = \arg\max_a Q(s,a)$

In most practical applications, ϵ is designed as a decreasing function of time t, implying that the probability of choosing an action randomly decreases as time progresses. This approach aims to prevent over-exploration. For instance, in the seminal paper discussing the convergence of Q-

learning to a collusive policy (Calvano, Calzolari, Denicolo, and Pastorello (2020)), the authors utilized an exponentially declining exploration rate over time:

$$\varepsilon_t = e^{-\beta t}. \tag{2.2}$$

Q-learning is a fundamental algorithm in the realm of reinforcement learning, and numerous advanced algorithms have been developed based on its core principles. For example, DQN (Deep Q-Network) (Mnih et al. (2015)) maintains a Q-function for each state-action pair, updating it not through the Bellman equation, but by employing a deep neural network to predict Q-values. Algorithms that make decisions based on state-action pair Q-functions are termed *value-based* algorithms. Currently, state-of-the-art reinforcement learning algorithms do not rely on the Q-function to estimate the expected payoff for each state-action pair. Instead, they directly parameterize and differentiate the policy. A notable example of such an algorithm is the Deep Deterministic Policy Gradient (DDPG) algorithm (Lillicrap et al. (2015)). For a comprehensive understanding of these advanced reinforcement learning algorithms, readers should refer to Sutton and Barto (2018).

The structure of dynamic pricing aligns closely with the Markov Decision Process (MDP) framework, where the state can be defined as the historical prices, the action can be the price, and the reward can be the profit. Past research has investigated the application of RL algorithms in dynamic pricing. For instance, Liu, Zhang, Wang, Deng, and Wu (2019) employs field experiments and shows that dynamic pricing strategies based on deep reinforcement learning can significantly outperform manual pricing approaches conducted by operational experts.

bandit learning is a special RL algorithm with no state space. It maintains an estimated reward for each action using the empirical mean and a penalty term to prevent frequent selections on sub-optimal action. For example, the famous Upper Confidence Bound (UCB) algorithm estimates the reward using the following representation:

$$UCB_i(t-1,\delta) = \underbrace{\frac{1}{N_{i,t-1}} \sum_{t' \le t-1} r_{t'} \mathbb{I}\{a_{t'} = i\}}_{\text{Empirical Mean}} + \underbrace{\sqrt{\frac{2\log(1/\delta)}{N_{i,t-1}}}}_{\text{Penalty Term}},$$

where δ is the confidence level and $N_{i,t-1}$ is the number of times action i has been chosen by time t. The empirical mean serves as the exploitation, while the penalty term is a way for exploration.

Prior research has integrated bandit learning into dynamic pricing strategies (Misra, Schwartz, and Abernethy (2019)).

Competitive dynamic pricing involves multiple agents who make decisions in the same environment. The literature on algorithmic collusion studies situations where all agents price **independently**, meaning each agent considers only its own actions. The algorithms illustrated above are all independent algorithms and are appropriate for such settings. There was research on multi-agent reinforcement learning, including multi-agent Q-learning (Hu, Wellman, et al. (1998)) and Nash Q-learning (Hu and Wellman (2003))[5]. However, these multi-agent reinforcement learning algorithms, which require agents to make decisions for all agents and require information about other agents, are not applied in the context of dynamic pricing.

2.2. Numerical Experiments of Algorithmic Collusion under Reinforcement Learning

Calvano, Calzolari, Denicolo, and Pastorello (2020) is the first paper discussing algorithmic collusion under reinforcement learning. Considering a market of n firms, each sells one differentiated product, the demand for firm i at time t is given by:

$$q_{i,t} = \frac{e^{\frac{a_i - p_{i,t}}{\mu}}}{\sum_{j=1}^{n} e^{\frac{a_j - p_{j,t}}{\mu}} + e^{\frac{a_0}{\mu}}}. \tag{2.3}$$

The parameter a_i is the vertical differentiation for each firm, a_0 is the utility of an outside good, μ is the index of horizontal differentiation, and $p_{i,t}$ is the price of product i at time t. The payoff function is given by $\pi_{i,t} = (p_{i,t} - c_i)q_{i,t}$, where c_i is the cost. There are two important prices in all papers on algorithmic collusion:

1. The collusive prices $\mathbf{p}^{col}$, also known as the monopoly prices, are calculated by maximizing the sum of profits of all firms;
2. The competitive prices $\mathbf{p}^{com}$ is the Bertrand-Nash equilibrium prices of the one-shot game.

It is assumed that $p_i^{com} < p_i^{col}$. The authors discretize the action space by limiting the feasible prices for each firm i to m equidistant points within the interval $[p_i^{com} - \xi(p_i^{col} - p_i^{com}), p_i^{com} + \xi(p_i^{col} -$

[5] The reason for mentioning independent algorithms here is that it has been proven that multi-agent Q-learning converges to the Nash equilibrium in any general-sum game. However, these algorithms are not Q-learning algorithms used in algorithmic collusion literature.

p_i^{com})], where $\xi > 0$ is a given parameter. Under the assumption that competitors' historical prices are publicly available information, each agent utilizes the prices from the last k periods of all agents to construct the state space:

$$s_t = \{\mathbf{p}_{t-1}, \ldots, \mathbf{p}_{t-k}\}$$

The style of constructing the action space has been succeeded by subsequent research. It is assumed that each firm is unaware of the structure of the demand function and employs the Q-learning algorithm, as described in Algorithm 1, to determine prices, with the only variation being in the termination condition. In each simulation, the game will terminate either upon convergence or after a maximum of two billion epochs, where **convergence** is defined as the situation where the optimal strategy for all agents remains unchanged over a consecutive 100,000 epochs. Upon completion of a simulation, the **average profit gain** Δ will be computed using:

$$\Delta \equiv \frac{\bar{\pi} - \pi^{com}}{\pi^{col} - \pi^{com}}, \tag{2.4}$$

where $\bar{\pi}$ represents the average sum of profits across all epochs, and π^{com} denotes the sum of profits when the market operates at competitive prices. When $\Delta = 0$, the game is characterized by full competition. As Δ increases, the outcome is termed *supra-competitive*, indicating that the average price set by the firms exceeds the Nash equilibrium. A Δ value of 1 signifies that both players consistently charge the collusive prices.

The size of the state space is given by $|S| = m^{nk}$, which grows exponentially with both n and k. To avoid memory issues, the authors restrict $n = 2$ and $k = 1$, implying that the market consists of only two firms, and each firm decides solely on the price from the previous epoch. The authors analyzed the average profit gain over various values of α, the learning rate, and β, the exploration parameter[6]. For each parameter combination, the authors conducted 1,000 simulations, and the results are presented in Figure 1.

[6] In the baseline simulation, the demand function does not incorporate stochasticity. Randomness and asymmetry are considered in their robustness checks, and the experiment exhibits similar results.

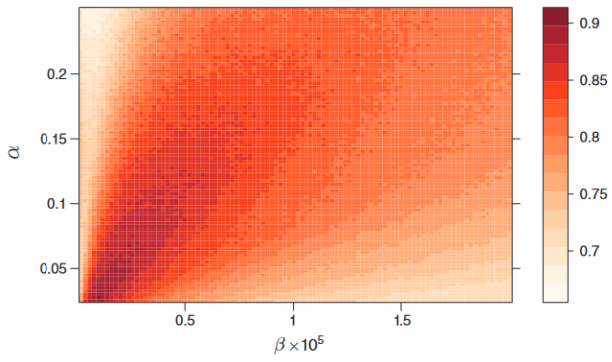

Figure 1: Δ For a Grid of Values of α and β

The baseline simulation involve a 100×100 parameter grid bwtween the learning rate α and the exploration parameter β, the shading of the color indicates the average Δ in 1,000 simulations for each parameter combination. This is copied from figure 1 in Calvano, Calzolari, Denicolo, and Pastorello (2020).

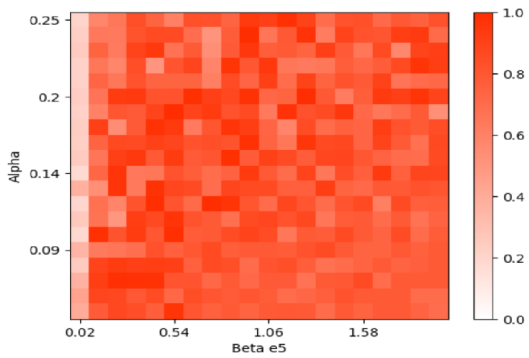

Figure 2: Replication for Figure 1

To reduce the replication time, a parameter grid with size 20×20 was used, and only 10 simulations were conducted for each parameter combination. The code generating this figure is available here.

Figure 2 represents my replication of a smaller scale. It is evident that in the majority of settings, Δ is significantly greater than 0, suggesting that the average profits are higher than the competitive profits. In certain settings, Δ nears 1, implying that the game has converged to a full monopoly situation. Given that the only information exchanged between the two firms is the historical prices, this situation can be described as a tacit collusion. However, it remains to be investigated whether this collusion can be classified as a *genuine collusion*—one that includes a punishment scheme for deviations—or if it remains a *spurious collusion*.

The authors explore this question by manually adjusting the prices of one firm after convergence has been achieved. Specifically, when the game has converged at time 0, the researchers randomly select one firm to be the *deviating agent*. At time 1, they set the price of the deviating agent lower than the converged price, ensuring that the deviating firm can earn more than by charging the supra-competitive price in a single-period game. Meanwhile, the non-deviating firm continues to set its price using the Q-learning algorithm. After time 1, both firms resume using the Q-learning algorithm to determine their prices, allowing the researchers to observe the pricing dynamics of the game. The results are shown in figure 3.

We can observe that at time 2, immediately following the forced deviation, the non-deviating agent initiates a price war by reducing its prices. Subsequently, both firms gradually increase their prices and return to the converged prices within 10 epochs. It should be noted that in more than 95% simulations, the punishment scheme results in the deviating agent being less profitable than if it had continued charging the supra-competitive prices. These findings show that the use of Q-learning for pricing can lead to *genuine collusion*. Calvano's study has had a significant impact,

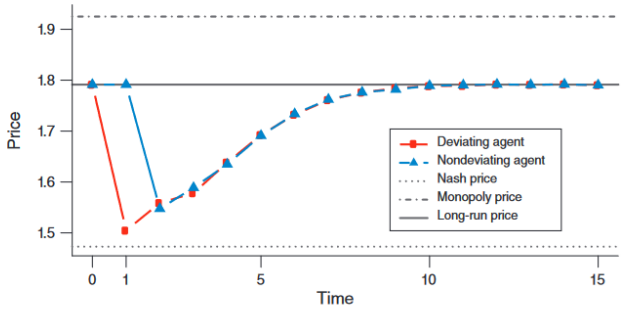

Figure 3
The figure illustrates the dynamics of prices following a forced deviation, with the data representing the average prices across the 1,000 sessions. This is copied from Figure 4 in Calvano, Calzolari, Denicolo, and Pastorello (2020).

particularly in the fields of industrial organization and game theory. Shortly after its publication, numerous studies have embarked on numerical experiments to investigate whether other reinforcement learning algorithms can learn to collude in this or different economic settings.

Hettich (2021) investigates whether DQN (Deep Q-Network) can learn to collude within Calvano's environment. The simulation indicates that DQN can learn to collude in a duopoly market and converge to supra-competitive outcomes more rapidly than the Q-learning algorithm, and the punishment scheme is also observed. DQN allows for testing in oligopoly markets with many agents since DQN does not require a large Q-matrix for every state-action pair. Simulations with multiple firms reveal that an increasing number of firms decreases Δ, and when $n \geq 10$, the convergent prices align exactly with the Nash equilibrium of a one-shot game (See figure 4).

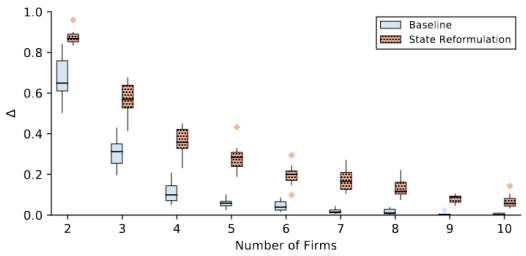

Figure 4
The figure depicts how Δ varies as the number of firms in the system increases. The *State Reformulation* box adjusts the state space to represent the average price from the last epoch, rather than the individual prices of both players. This is copied from figure 6 in Hettich (2021).

In addition to the economic environment employed in Calvano's work, many researchers are focusing on the Bertrand pricing model. In the Bertrand model, two firms sell a homogeneous product, and consumers choose the product with the lowest price in the duopoly market. The

demand function for firm i is given by:

$$d_i(p_i, p_j) = \begin{cases} D(p_i), & \text{if } p_i < p_j, \\ D(p_i)/2, & \text{if } p_j = p_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

Asker, Fershtman, and Pakes (2022), Asker, Fershtman, and Pakes (2023) explores state-free Q-learning algorithms within the Bertrand model. Several features of this study are highlighted below:

1. The state space for the Q-learning algorithm is reduced to a singleton, indicating that the algorithm does not rely on historical prices;
2. $\delta = 0$, implying that the algorithm considers only the one-period payoff;
3. There is no ϵ exploration; instead, at each epoch, the agent persistently uses the greedy action;
4. The authors term the Q-value updating scheme in algorithm 1 as the *asynchronous* version of Q-learning and introduce another updating scheme known as *synchronous* updating[7], which is detailed in algorithm 2.

Algorithm 2 Synchronous Algorithm in Asker et al. (2022)

Initialize action-value function $Q(a_i)$ for all a_i in ($\mathcal{A}$
Input a counterfactual reward-estimation function $r^e(\cdot)$ and initialize a learning rate α
$s \leftarrow$ initial state
while Not Converge **do**
    $a_i \leftarrow \mathbf{argmax}_{a_i} Q(a_i, a_j)$
    Take action a_i, observe opponent's action a_j and reward $r_i(a_i, a_j)$
    $Q(a_i) \leftarrow \alpha r_i(a_i) + (1-\alpha)Q(a_i)$
    $Q(a_k) \leftarrow \alpha r^e(a_k, a_j) + (1-\alpha)Q(a_k)$ for all $k \neq i$ ▷ Synchronous Updating
end while

The numerical results indicate that the state-free Q-learning algorithm that doesn't care about future payoffs ($\delta = 0$) can converge to supra-competitive outcomes. Even when both firms use a

[7] The synchronous updating scheme requires the input of a counterfactual reward-estimation function $r^e(\cdot)$, which is utilized to estimate $r_i(a_i, a_j)$. Synchronous updating requires that when action a_i is chosen in a given epoch, all corresponding Q-values $Q_i(a_i)$ must be updated. In the ideal but unrealistic case of perfect synchronous updating, the firm is assumed to employ the true demand function as $r^e(\cdot)$. The term *synchronous* as used here deviates from the convention in computer science literature, and I personally maintain a cautious stance regarding the practical application of this algorithm.

downward-sloping demand function as the counterfactual reward-estimation function to update their Q-functions symmetrically, the game tends to result in supra-competitive pricing. The game can converge to the Nash equilibrium only under the perfect synchronous updating scheme, in which the firms know the demand function.

Klein (2021) explores the sequential Bertrand pricing game, where firms adjust prices **sequentially**: Firm 1 can only adjust its price in odd-numbered periods, while Firm 2 adjusts its price in even-numbered periods. In the Markov Perfect Equilibrium, firms engage in undercutting each other by one increment until prices reach the lower bound, after which one firm resets prices to one increment above the monopoly price, and the cycle restarts. This equilibrium is called the Edgeworth price cycle (Maskin and Tirole (1988)). In the numerical testing, both firms use a modified version of Q-learning to price:

1. To satisfy the Markov assumption, the state space is defined as the competitor's previous price, rather than the previous price of both firms;
2. Since the game is sequential, the updating scheme of the Q-value is modified to:

$$Q(p_{i,t}, s_t) = \pi_i(p_{i,t}, s_t) + \delta\pi_i(p_{i,t}, s_{t+1}) + \delta^2 \max_p Q_i(p, s_{t+1})$$

The results show that when the action space is small ($k = 6$), the game can converge to the genuine collusion; when the action space is large ($k = 24$), the game tends to settle into a pattern similar to the Edgeworth cycle[8] (see Figure 5), yet the average profitability still exceeds that of the equilibrium.

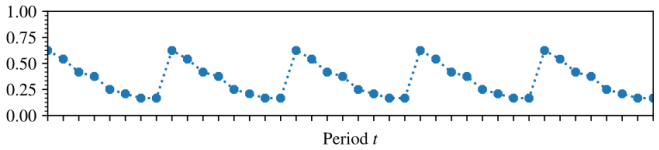

Figure 5
The figure illustrates the final average market price, which tends to settle into an Edgeworth cycle pattern: firms undercut the price of one increment until the price reaches their lower bound, then one firm resets price and the cycle restarts. This is copied from figure 7 in Klein (2021).

Bandit algorithm can still bring tacit collusion (Hansen et al. (2021)). Consider a symmetric duopoly linear demand:

$$\pi_{i,t} = (\alpha - \beta p_{i,t} + \gamma p_{-i,t}) \cdot p_{i,t} + \epsilon_t, \quad \epsilon_t \sim U[-\frac{1}{\delta}, \frac{1}{\delta}],$$

[8] However, although the dynamics look like the Edgeworth cycle, they are not the same.

where δ represents the signal-to-noise ratio (SNR) and a lower SNR results in greater profit stochasticity. Each firm utilizes the *UCB-tuned* algorithm to update the value for each bandit k at time t:

$$UCB-tuned_{k,t} = \underbrace{\bar{\pi}_{k,t}}_{\text{Empirical Mean}} + \underbrace{\sqrt{\frac{\log t}{n_{k,t}} min\left(\frac{1}{4}, \overline{\pi^2_{k,t}} - \bar{\pi}^2_{k,t} + \sqrt{\frac{2\log t}{n_{k,t}}}\right)}}_{\text{Penalty Term}}, \tag{2.6}$$

where $n_{k,t}$ is the represents the frequency of the agent taking action k up to time t, and $\overline{\pi^2_{k,t}} - \bar{\pi}^2_{k,t}$ is the empirical variance. The action space is limited to only two actions: p^{col} and p^{com}[9]. Figure 6 visualizes the density of convergent prices. As the SNR increases, making the profit function more stable, the game is more likely to converge to a monopoly outcome.

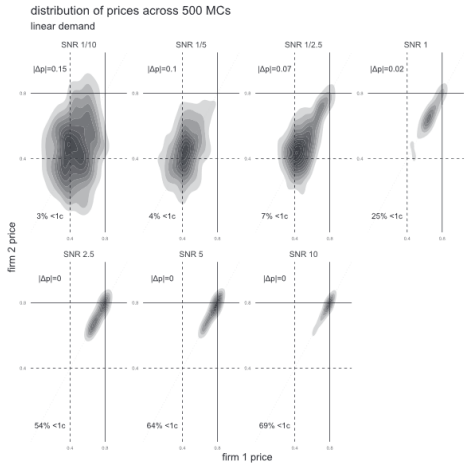

Figure 6: Game convergence under different SNR

In the baseline experimentation, $p^{com} = 0.4, p^{col} = 0.8$. The more the shadow leans towards the upper right corner, the higher the chance the game will converge to collusion. This is copied from figure 2 in Hansen et al. (2021).

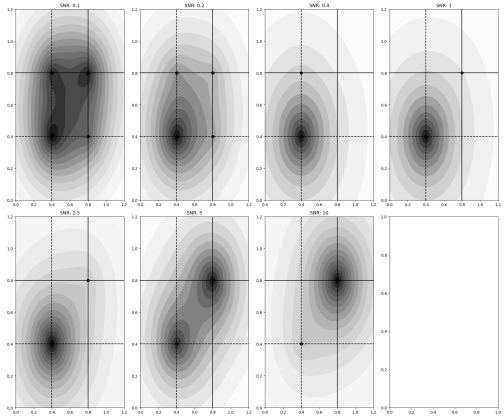

Figure 7: Replication for Figure 6

The parameter setting is exactly the same as the original work. The code generating this figure is available here.

Other numerical studies on algorithmic collusion by RL include Fish, Gonczarowski, and Shorrer (2024), which examines large language models; Martello (2022), which examines the actor-critic algorithm; and Koirala and Laine (2024), which examines the Proximal Policy Optimization algorithm. See Table 2 for a summary of all numerical research.

2.2.1. Historical Research on Algorithmic Collusion in Cournot Game

In fact, long before Calvano et al. (2019), research on algorithmic collusion in Cournot games had been conducted. In a Cournot game, multiple firms determine the production level (quantity)

[9] The competitive and collusive prices can be computed explicitly: $p^{com} = \frac{\alpha}{2\beta-\gamma}$, $p^{col} = \frac{\alpha}{2(\beta-\gamma)}$.

Algorithm	Environment	Collusion	Type of Collusion
Q-learning Calvano, Calzolari, Denicolo, and Pastorello (2020)	Calvano	Yes	Genuine
DQN Hettich (2021)	Calvano	Yes	Genuine
Actor-Critic Martello (2022)	Calvano	Yes	Genuine
LLM Fish et al. (2024)	Calvano	Yes	Unkown
Perfect Synchronous State-Free Q-learning Asker et al. (2022)	Bertrand	No	/
Asynchronous State-Free Q-learning Asker et al. (2023)	Bertrand	Yes	Spurious
Sequential Q-learning Klein (2021)	Sequential Bertrand	Yes	Unkown
UCB-tuned Hansen et al. (2021)	Linear Duopoly	Yes	Unkown

Table 2 A Summary of Numerical Studies of Algorithmic Collusion under RL

of products, and the market price is determined by the total production levels. Kimbrough, Lu, and Murphy (2005) consider the duopoly Cournot game where both agents employ a state-free Q-learning algorithm to select actions. Rather than using a static strategy, the study utilizes five **molecular strategies**. Denote the production quantity of one firm at time t as x_t and that of the opposing firm as y_t:

- *Generous Tit for Tat*: if $y_{t-1} > y_{t-2}$, then $x_t = x_{t-1} + \delta$; else $x_t = x_{t-1} - \delta$;
- ...
- *BestResponse*: the best response to y_{t-1} given the price function.

Both firms use the *BestResponse* strategy is the SPNE with the lowest profit. The average profit under Q-learning exceeds the lowest profit, suggesting that Q-learning results in tacit collusion in this game. Further research into tacit collusion facilitated by AI algorithms can be found in Waltman and Kaymak (2008), Kimbrough and Murphy (2009), and Siallagan, Deguchi, and Ichikawa (2013).

2.3. Criticize of Algorithmic Collusion in Numerical Experiments

Many voices argue that algorithmic collusion is far more challenging in practice (Schwalbe (2018)). The main points of criticism can be summarized into four aspects:

Firstly, while collusion may appear to involve a reward-punishment scheme, it is not sufficiently smart to pose a credible threat to an opponent's deviation. It is widely believed that algorithmic collusion results from the insufficient exploration inherent in the Q-learning algorithm. For example, Abada and Lambin (2023) test the case where agents were forced to explore randomly with a probability double that of the standard probability. They found that the market dynamics were closer to the perfect competition under the modified exploration mode. Another piece of evidence indicating imperfect exploration as a cause comes from figures 1 and 2. Note that the leftmost side of the graphs, where the algorithm focuses more on exploration, is accompanied by a smaller Δ. This suggests that the Nash equilibrium may be achieved when the Q-learning is given enough time to explore.

What's more, the algorithms can sometimes behave extremely irrationally. Figure 3 seems to depict a perfect retaliation scheme, but it aggregates the dynamics of 1,000 rounds. 8 depicts the trace of one single experiment. The traces are quite disorganized, and the converging prices are not identical for the two firms, even though the parameter setting is symmetric.

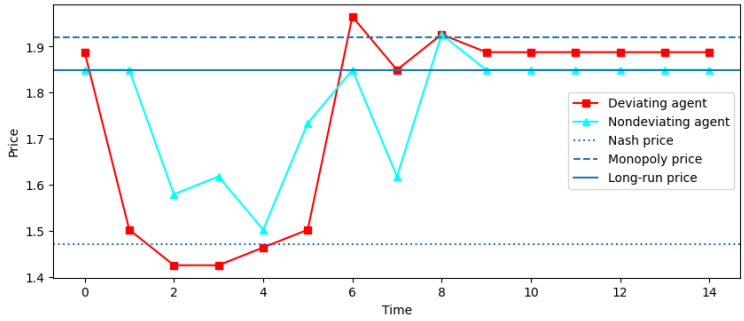

Figure 8
At time 1, the deviating agent is forced to set a low price, while the non-deviating agent continues to price using the algorithm. Thereafter, the prices for both agents are determined exclusively by the algorithm.

Additionally, Epivent and Lambin (2024) shows that the reward-punishment scheme does not respond solely to deviation behaviors when an opponent decreases price; instead, it occurs whenever the opponent adjusts their prices, even when the opponent swifts to a higher price. Thus, many scholars argue that the supra-competitive convergence, although static for more than 100,000 epochs in the simulation, is not any well-defined equilibrium i.e. *"what looks like collusion need not be collusion"* (den Boer, Meylahn, and Schinkel (2022)). Without theoretical support, the market may trend closer to competitive outcomes after sufficient exploration.

Secondly, the *number effect* (Schwalbe (2018)) should be considered in real markets. This effect suggests that as the number of firms participating in the market increases, the tacit collusion

induced by Q-learning may be mitigated or even eliminated. This phenomenon has been verified by several studies including Hettich (2021), Asker et al. (2022), Abada and Lambin (2023).

Thirdly, while numerical experiments can model aspects of the real world, the current simulations are still quite distinguished from the complexities of actual environments. For example, it's not likely for firms to adopt the Q-learning algorithm to price when it can be outperformed by other simpler algorithms, such as Exp3 (den Boer et al. (2022)).

Finally, the conditions for the existence of a collusive equilibrium in the numerical experiment's environment are harsh. den Boer et al. (2022) refined the definition of equilibrium to accommodate the dynamics of dynamic pricing using ϵ-greedy Q-learning in a duopoly market. Similar to previous research, the state space is defined as the actions of both firms in the last epoch, i.e., $s \in \mathcal{A}_{-i} \times \mathcal{A}_i$.

Definition 1. $(\delta-\epsilon)$-*best-response*: A strategy $\sigma^{(i)} = \{\sigma^{(i)}(s) : s \in \mathcal{A}_i \times \mathcal{A}_{-i}\}$ of player i is called $(\delta-\epsilon)$-*best-response* to strategy $\sigma^{(-i)} = \{\sigma^{(-i)}(s) : s \in \mathcal{A}_{-i} \times \mathcal{A}_i\}$ if the following eqution holds for all states s and s_{-i}:

$$\begin{aligned}\sigma^{(i)}(s) \in \arg\max_{a\in\mathcal{A}_i} &\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{a_{-i}\in\mathcal{A}_{-i}}\{r_i(a,a_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(a,a_{-i})\} \\ &+ (1-\epsilon)\{r_i(a,\sigma^{(-i)}(s_{-i},s_i)) + \delta V^{(i)}_{\sigma^{(-i)}}(a,\sigma^{(-i)}(s_{-i},s_i))\},\end{aligned} \tag{2.7}$$

where the value function $V^{(i)}_{\sigma(-i)}(s)$ is defined by:

$$\begin{aligned}V^{(i)}_{\sigma(-i)}(s) := \max_{p\in\mathcal{A}_i} &\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{a_{-i}\in\mathcal{A}_{-i}}\{r_i(a,a_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(a,a_{-i})\} \\ &+ (1-\epsilon)\{r_i(a,\sigma^{(-i)}(s_{-i},s_i)) + \delta V^{(i)}_{\sigma^{(-i)}}(a,\sigma^{(-i)}(s_{-i},s_i))\},\end{aligned} \tag{2.8}$$

A strtegy pair $(\sigma^{(i)},\sigma^{(-i)})$ is claled a $(\delta-\epsilon)$-*strategy-equilibrium* if they are mutually $(\delta-\epsilon)$-best-response to each other.

This definition of best response aligns with the exploratory nature of Q-learning. When both firms use ϵ-greedy Q-learning to make decisions, the optimal strategy and equilibrium must account for the opponent's random exploration behavior. It has been proven that for any δ, there exists an $\epsilon^*(\delta) \in (0,1)$ such that for all $\epsilon > \epsilon^*(\delta)$, the only $(\delta-\epsilon)$-strategy-equilibrium is both firms use the competitive price (for further details, see den Boer et al. (2022)). This implies that the conditions for the existence of a theoretically reliable equilibrium are stringent, even if the simulation environment were to be applied in the real world.

2.4. Theoretical Evidence of Algorithmic Collusion

Den Boer's definition of $(\delta - \epsilon)$-strategy-equilibrium greatly can explain the slow convergence observed in Calvano's numerical experiments. The supra-competitive $(\delta-\epsilon)$-strategy-equilibrium only exists when ϵ, the exploration rate, is sufficiently small. In Q-learning, ϵ decreases over the number of rounds, and it can take millions of rounds to reach a small enough ϵ for the collusive equilibrium to occur.

However, this framework does not explain the collusive convergence in some settings. For example, Asker et al. (2023) examined a scenario where $\delta = 0$, in which the theoretically sustainable collusive equilibrium should not exist, yet numerical results still converge to a supra-competitive outcome. This suggests that the theory presented in den Boer et al. (2022) does not provide a full picture of the dynamics of Q-learning in pricing problems.

To date, there is **no universal** conclusion on the cause of algorithmic collusion by the reinforcement learning algorithm. While ample evidence suggests that both genuine and spurious collusion are driven by imperfect exploration, there remains a lack of analytical proof to support this assertion. Here, I survey some research that seeks to address the fundamental question: why can Q-learning, along with other reinforcement learning algorithms, lead to collusive convergence in an oligopoly market?

Banchio and Mantegazza (2022) proposes that algorithmic collusion relies on an endogenous statistical linkage within the Q-values, which they term *spontaneous coupling*. Spontaneous coupling arises from the correlation of Q-values due to the updation being dependent on both firms' actions. When the algorithm estimates the Q-value for the collusive price, it actually estimates the Q-value for the collusive price, conditioning the opponent using a collusive price. Conversely, the estimated Q-value for the competitive price is conditioned on the opponent using the competitive price. This can lead to an overestimation of the Q-value for higher prices. The paper presents sufficient conditions when spontaneous coupling does not occur[10]: under synchronous updating Q-learning, coupling does not occur due to the uniform learning rate across different actions. This theorem aligns with the numerical results presented in Asker et al. (2022), which found that perfect synchronous Q-learning leads to a Nash equilibrium.

[10] see figure 6 in Banchio and Skrzypacz (2022). The existence of a collusive equilibrium is contingent on both the payoff function and the exploration rate; typically, a higher exploration rate reduces the likelihood of a collusive equilibrium forming.

Dolgopolov (2024) employs evolutionary game theory to analyze the convergence of **state-free** Q-learning agents in a prisoner's dilemma game. The theoretical analysis indicates that:[11]:

1. In the absence of exploration, state-free Q-learning would converge to the static Nash equilibrium;
2. When both agents use logit (Boltzmann) exploration, the convergence is contingent upon a closed-form relationship between the learning rate and the payoffs (see figure 9).

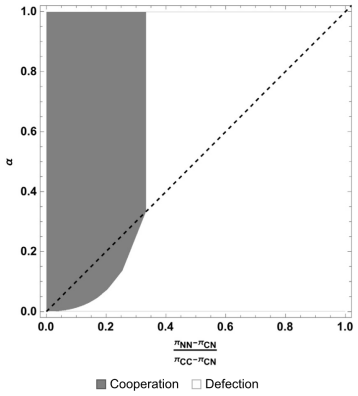

Figure 9
The figure depicts the relationship between the learning rate and the payoffs, with α representing the learning rate, π_{CC} being the payoff when both firms collude, π_{NN} being the payoff when both firms use the Nash price, and π_{CN} being the payoff when one firm tries to collude while the other deviates. This is copied from figure 2 in Dolgopolov (2024).

The first statement contradicts the numerical experiments in Asker et al. (2022), where firms in a Bertrand model employed greedy state-free Q-learning but still charged supra-competitive prices. This suggests that the analytical results from Dolgopolov (2024), based on a simple prisoner's dilemma game with a limited action space, may not be universally applicable. Minor variations in the environment can significantly alter the generality of a theorem.

The aforementioned theoretical research primarily concentrates on toy models with symmetric parameters for competing firms. Brown and MacKay (2023) examine the asymmetric setting where firms can price based on an opponent's most recent price and find that the price level will be supra-competitive under the MPE. The research above has collectively concluded that, under specific circumstances and parameter settings, the use of simple Q-learning or other algorithms that condition on an opponent's previous price can lead the game to converge to a collusive scenario, with some of these convergences representing well-defined equilibrium.

[11] See proposition 3, corollary 4 and 5 from Dolgopolov (2024) for more details.

2.4.1. Theoretical Evidence beyond Reinforcement Learning

Some studies have provided analytical proof of collusion potential under other algorithms beyond the scope of reinforcement learning. Aouad and den Boer (2021) demonstrated that even a simple ϵ-greedy algorithm can lead a duopoly assortment game converging to a collusive outcome, rather than the equilibrium in the assortment game (Besbes and Sauré (2016)). Cho and Williams (2024) proved that firms using a simple model averaging and least squares estimation to price converge to a collusive scenario in a linear Bertrand duopoly setting. What's more, the convergence is not contingent on any channel including the patient-player assumption, suggesting that the algorithm can lead to higher prices even without a reward-punishment scheme.

Another study of algorithmic collusion that has been rigorously established is presented in Lamba and Zhuk (2022). In this model, two firms are selling a homogeneous product within the same market, with each firm dynamically pricing their goods solely based on the opponent's price during the last epoch. The pricing strategies are distilled to a few, including always using the monopoly price, always using the collusive price, tit-for-tat, and reverse tit-for-tat. The authors have proved that under the Markov Perfect Equilibrium, the average price set by both firms is supra-competitive.

Some insights can also be drawn from studies of collusion in Cournot games such as Shi and Zhang (2020), which demonstrated that the policy-gradient algorithm can converge to the Nash equilibrium in a Cournot game under certain conditions, including when the price function is linear or when the market is duopolistic. Additionally, Possnig (2023) derived sufficient conditions for actor-critic algorithms to learn collusion in a duopoly Cournot model.

2.5. Empirical Evidence of Algorithmic Collusion

The most compelling empirical evidence comes from Clark, Assad, Ershov, and Xu (2023). The authors utilize price data from the German gasoline market in this study. They focus on a specific market structure: the duopoly market, where two gas stations operate within a nearby region. They identify the timing when each gas station starts to use algorithms to price and conduct the following regression:

$$y_{mt} = \alpha_m + \alpha_t + \beta_1 T^1_{mt} + \beta_2 T^2_{mt} + \epsilon_{mt},$$

where y_{mt} represents the price margin in market m at time t, and T^1_{mt} and T^2_{mt} are dummy variables that indicate whether market m at time t has only one or two gas stations using algorithms to price, respectively. Using instrumental variables, the authors identify that in duopoly markets, the adoption of algorithmic pricing by a single firm does not significantly affect the price margin, whereas the adoption by both firms can increase station-level margins by 28%[12].

Wieting and Sapi (2021) shows the evidence of algorithmic collusion intuitively, figure 10 depicts the correlation coefficient between the market price and the number of firms implementing algorithmic pricing. In a duopoly market ($comp = 2$), the price where two firms use algorithms is significantly higher than when only one firm uses algorithms, suggesting algorithmic collusion.

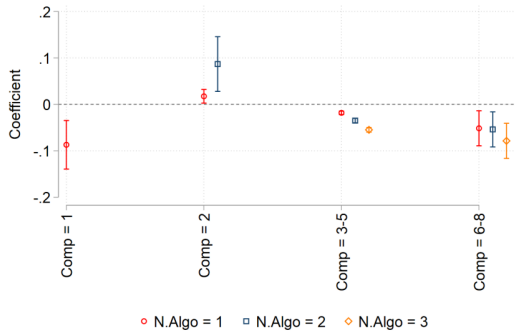

Figure 10
This picture shows the correlation coefficient between the price and the number of firms using algorithms. *Comp* indicates the number of competing firms in each cluster, *N.Algo* is the number of firms using algorithms. This is copied from figure 11 in Wieting and Sapi (2021).

Another empirical evidence comes from Musolff (2022), which reveals that the mean price in Amazon's marketplace tends toward the monopoly price under algorithmic pricing. However, all empirical studies only indicate that algorithmic collusion can occur, without specifying that the algorithms used are necessarily RL algorithms. The inner mechanisms behind such tacit collusion remain opaque or as a 'black box.'[13]

[12] Furthermore, Clark et al. (2023) provide evidence that margins do not increase until about a year after market-wide adoption, suggesting that the price increase is due to the algorithm's active learning on how to avoid competition. This indicates that the algorithmic collusion observed in the gasoline market is genuine rather than spurious.

[13] some other research indicates that algorithmic pricing may not solely result in price increases. For instance, Calder-Wang and Kim (2023) has identified that algorithms can set more responsive prices i.e. the prices set by algorithms can more effectively adapt to dynamic demand.

2.6. Concluding Remarks on Algorithmic Collusion under RL

In this part, I will offer a brief conclusion based on my own reflections.

Many numerical studies have shown that using reinforcement learning algorithms to set prices in competitive markets can lead to supra-competitive pricing, often supported by a reward-punishment scheme that sustains collusion. These numerical findings have also been supported by a range of theoretical and empirical evidence, although not exhaustive. Theoretical results support that under certain conditions[14], using RL algorithm to pricing can converge to the collusive outcome. Although these theories are primarily based on the prisoner's dilemma game, their insights align with the results of numerical experiments conducted in more complex settings[15]. On the other hand, empirical evidence has identified that using algorithms to price, which may not necessarily involve RL algorithms, can raise the market price in a competitive market.

These numerical, analytical, and empirical evidence, while not complete, reveal that algorithmic collusion is a problem that deserves consideration. Critics may argue that the collusion, including the reward-punishment scheme, is not a well-defined equilibrium and that simple reinforcement learning algorithms can be outperformed by others, making RL algorithms a *dominated strategy* for pricing. My perspective is: algorithmic collusion (including both spurious and genuine collusion), does not require a complete theoretical framework to justify its existence. It is observed to occur in both simulations and the real market.

The convergence of a pricing game under RL algorithm does not need to be a theoretically well-defined equilibrium.

For example, most of the numerical studies in Asker et al. (2022) assume $\delta = 0$, suggesting that there is no room for any collusive equilibrium to exist. However, Q-learning can still converge to supra-competitive outcomes in some numerical experiments. Addressing the incomplete nature of algorithmic collusion under RL is a meaningful discussion, but this does not negate the significance of algorithmic collusion under RL algorithms. This is why I summarize the three types of algorithmic collusion in Section 1.

[14] Depending on parameters such as the payoff function, learning rate, exploration rate, and Q-value updating scheme.

[15] For instance, Theorem 3 from Banchio and Skrzypacz (2022) concurs with the numerical findings in Asker et al. (2023) when employing a perfect synchronous updating scheme. Figure 1 from den Boer et al. (2022) sheds light on why, in Abada and Lambin (2023), when the authors doubled the exploration probability every epoch, social welfare increased.

However, these critiques have raised a significant question: Is there an algorithm that can learn to collude with a theoretical guarantee and without any mechanism that could break? This question introduces the concept of *authentic collusion*, which I will elaborate on in the subsequent section.

3. Authentic Algorithmic Collusion

All algorithmic collusions in Section 2, whether exhibiting a reward-punishment scheme, fall under the category of spurious or genuine collusion. This is because other pricing strategies may outperform the RL algorithm, exploiting unobservable weaknesses to gain a higher cumulative payoff. The core of authentic algorithmic collusion lies not only in the analytical guarantee of convergence to a supra-competitive outcome, but also in its resilience against other pricing algorithms.

3.1. A Rigorous Definition of Algorithmic Collusion

den Boer (2023) offers a fantastic definition of algorithmic collusion for the duopoly market. Here, I provide a summary that encapsulates the essence of this definition. Consider a repeated game between two firms, denoted by $i = 1$ and $i = -1$, where both firms are selling a homogeneous product. At each discrete time step $t \in \mathbb{N}$, each firm selects a price P_{it} from a set of feasible prices $\mathcal{P}_i$. The demand D_{it}, D_{-it} obtained by firms at time t is drawn from the **demand function** $d(\mathbf{P}_t)$, where $\mathbf{P}_t$ is the vector of prices. The demand function is assumed to be independent of the history of the game, meaning that for a given price pair $\mathbf{P}$, the demand D_i and D_{-i} follow the distribution $d(\mathbf{P})$ that is constant over time. After observing the demand D_{it}, each firm receives a revenue $R_{it} = P_{it} \cdot D_{it}$. The structure of this game is denoted by the tuple $(\mathcal{P}_1, \mathcal{P}_{-1}, d)$.

Most research assumes that prices are publicly known, while the demand D_{it} remains private information[16]. The demand function d is unknown to the firms, who must learn the market environment by setting prices using algorithms to interact with their competitors.

The algorithm (pricing strategy) $\pi_i \in \Pi_i$ is defined as the mapping from the *state* $H_{it} := (P_{is}, P_{-is}, D_{is} \text{ for } 1 \leq s \leq t)$ to the probability distribution over the action space $\mathbb{P}(\mathcal{P}_i)$. Given that

[16] There is an expectation, see Meylahn and V. den Boer (2022).

the game is solely dependent on $d(\cdot)$ and the algorithms used by firms, the expected cumulative reward for each firm can be denoted as:

$$\phi_i(\pi_i, \pi_{-i}, d)$$

It's untractable to compute the best response against an algorithm, so the *regret* of an algorithm is defined as the difference from the reward under the **optimal fixed** action given the opponent's algorithm π_{-i} and the demand function $d(\cdot)$.

$$Regret_i(\pi_i, \pi_{-i}, d) := \sup_{P_i^* \in \mathcal{P}_i} \phi_i(P_i^*, \pi_{-i}, d) - \phi_i(\pi_i, \pi_{-i}, d)$$

Now, we can formally define algorithmic collusion[17]. We say a pair of algorithms $\tilde{\pi}_i, \tilde{\pi}_{-i}$ can lead to authentic algorithmic collusion under the demand function $d(\cdot)$ if they satisfy:

1. The algorithm, as described earlier, is designed to operate solely based on historical prices and demands, implicitly assuming that there is no illegal communication between the two firms.
2. Algorithmic collusion needs to achieve supra-competitive payoffs. We denote $\phi_i^{comp}(d)$ as the expected payoff during a fully competitive setting, which is typically defined as the payoff under one-stage Nash Equilibrium prices. For each firm $i = \pm 1$:

$$\phi_i(\tilde{\pi}_i, \tilde{\pi}_{-i}, d, f) \geq \phi_i^{comp}(d)$$

3. A 'smart' collusion must have a good enough performance against alternative algorithms. For any $\pi_{-i} \in \Pi_{-i}/\tilde{\pi}_{-i}$ and some $\epsilon \geq 0$[18], the regret of the collusive algorithm pair must satisfy:

$$Regret_i(\tilde{\pi}_i, \pi_{-i}, d) \leq \epsilon$$

4. Finally, based on the preceding requirements, a collusion can be considered **authentic collusion**[19] if it meets an additional criterion: given that the opponent employs the collusive algorithm $\tilde{\pi}_{-i}$, the best response is to also use the collusive algorithm $\tilde{\pi}_i$. For each firm $i = \pm 1$, and for any $\pi_i \in \Pi_i/\tilde{\pi}_i$:

$$\phi_i(\tilde{\pi}_i, \tilde{\pi}_{-i}, d) \geq \phi_i(\pi_i, \tilde{\pi}_{-i}, d)$$

[17] In this survey, the framework of the pricing game is akin to that presented in the original work den Boer (2023). However, to help readers understand the connections between various types of collusion, there is a notable difference in the definition of authentic algorithmic collusion from the original text. den Boer (2023) treats the collusive and alternative algorithms as sets of algorithms and provides a more precise definition.

[18] ϵ can be regarded as a pre-determined parameter indicating the performance requirement.

[19] In den Boer (2023), a further requirement that the collusion does not assume synchronization is not emphasized in this paper.

It is crucial to note that the reward-punishment scheme presented in Section 2 can not satisfy the third requirement. Because the opponent's deviation, as described in previous literature, is only a subset of alternative algorithms. A huge limitation of this survey is that this framework does not intuitively distinguish between genuine collusion and authentic collusion, because it's challenging to define deviation using rigorous notations.

In summary, the hierarchy of the three types of algorithmic collusion mentioned in Section 1 can be more intuitively characterized as follows:

1. *Spurious collusion*: No illegal communication but can achieve supra-competitive outcomes;
2. *Genuine collusion*: Based on the preceding requirements, the collusive algorithm should exhibit a reward-punishment scheme toward deviation (a subset of alternative algorithms);
3. *Authentic collusion*: Based on the preceding requirements, the collusive algorithm should have low regret against all alternative algorithms and is the best response to the collusive algorithm.

3.2. Some Examples of Authentic Algorithmic Collusion

Currently, two papers have proposed authentic collusion algorithms[20].

Meylahn and V. den Boer (2022) introduce the first auto-collusive algorithm with guaranteed theoretical performance and convergence speed. The logic of this collusive algorithm is as follows: given the demand information is public, the firms can estimate the collusive prices and the revenue under the collusion, as well as the competitive prices and the revenue under the Nash equilibrium. The firm can then choose to collude or compete by determining which mode is more profitable. The algorithm will continue to perturb and update its estimations during the exploitation stage to mitigate regret in the event of deviations.

Following the notation introduced in Subsection 3.1, we can define the joint revenue function of the two firms as $r(\mathbf{P}_t) = \sum_{i\in\{1,-1\}} P_{i,t} \cdot d(\mathbf{P}_t))$. The authors impose certain assumptions[21] on the

[20] These two papers may be quite technical for the researcher in Econ/OM. I recommend that readers interested in this topic watch this presentation given by Arnoud den Boer.

[21] Refer to assumption 1 in Meylahn and V. den Boer (2022); these assumptions mainly entail that the collusive price pair is unique and that the regions nearing $\mathbf{P}^{col}$ exhibit sufficient steepness to ensure the convergence speed of the gradient algorithm.

joint revenue function r to enable the firms to employ gradient ascent algorithms to learn the collusive prices $\mathbf{P}^{col}$.

$$\mathbf{P}^{col} = \arg\max_{\mathbf{P}} r(\mathbf{P})$$

More specifically, each firm maintains an estimation $\hat{p}_i^{col}(n)$, where n is a counter. In each two consecutive time periods, the firm charges $\hat{p}_i^{col}(n) + c_n\omega_i^{col}(n)$ and $\hat{p}_i^{col}(n) - c_n\omega_i^{col}(n)$, where c_n is a tuning sequence and $\omega_i^{col}(n)$ is a random variable equals 1 and -1, both with probability $\frac{1}{2}$. After observing the demands, the joint revenue $\tilde{r}^+$ and $\tilde{r}^-$ during the two periods can be calculated, then each firm can obtain the gradient of $r(\cdot)$ using:

$$\hat{\nabla} r(n) := \frac{\tilde{r}^+ - \tilde{r}^-}{2c_n\omega_i^{col}(n)} \tag{3.1}$$

After that, the firms can update $\hat{p}_i^{col}(n)$:

$$\hat{p}_i^{col}(n+1) = \mathcal{P}_i\left(\hat{p}_i^{col}(n) + a_n\hat{\nabla} r(n)\right), \tag{3.2}$$

where $\mathcal{P}_i$ is a projection operator to bound $\hat{p}_i^{col}(n+1)$ and a_n is the learning rate sequence. This scheme is called Simultaneous Perturbation Kiefer–Wolfowitz recursions, which is a well-known method to stochastically estimate the maximum of a function.

It is important to note from equation 3.1 that to compute the gradient $\hat{\nabla} r(n)$, each firm requires knowledge of the opponent's demand information, which is a fundamental assumption in the algorithm. This assumption is plausible in some settings, as many e-commerce retailers now disclose their real-time inventory levels, thereby allowing competitors to infer demand at each interval.

Using the same methodology, the algorithm can employ the Simultaneous Perturbation Kiefer–Wolfowitz recursion to estimate $\hat{p}_i^{com}(n)$, which is the Nash equilibrium prices. The algorithm then determines whether setting $\hat{p}_i^{col}(n)$ or $\hat{p}_i^{com}(n)$ would be more profitable. Finally, the algorithm would transition to the exploitation stage. The complete algorithm is termed the composite SPSA (Simultaneous Perturbation Stochastic Approximation) algorithm, which splits the (infinite) time horizon into repeating cycles, and each cycle comprises five stages:

1. *Collusive exploration*: using the Simultaneous Perturbation Kiefer–Wolfowitz recursion to estimate $\hat{p}_i^{col}$;
2. *Collusive estimation*: set the price statically at $\hat{p}_i^{col}$, and obtain an accurate estimation of the revenue under the prices $\hat{\mathbf{p}}^{col}$;
3. *Competitive exploration*: using the Simultaneous Perturbation Kiefer–Wolfowitz recursion to estimate $\hat{p}_i^{com}$;
4. *Competitive estimation*: set the price statically at $\hat{p}_i^{com}$ to obtain an accurate estimation of the revenue under the prices $\hat{\mathbf{p}}^{com}$;
5. *Exploitation*: The firm will decide whether to use $\hat{p}_i^{col}$ or $\hat{p}_i^{com}$, and during this process, the Simultaneous Perturbation Kiefer–Wolfowitz recursion will continue to be employed.

The authors demonstrated that when both firms employ the composite SPSA algorithm, $\hat{p}_i^{col}$ and $\hat{p}_i^{com}$ will converge to p_i^{col} and p_i^{com}, respectively with a sublinear convergence rate. The market will converge to the collusive scenario if collusion is more profitable. What's more, the performance against the reaction function is guaranteed, and it represents the best response when the opponent uses the same algorithm, making it the first authentic collusive algorithm. The composite SPSA has two main drawbacks:

1. It requires knowledge of the mutual demand function.
2. The SPSA algorithm requires synchronization. i.e. if the two firms do not initiate the same algorithm simultaneously, they may not end up charging the collusive prices.

Loots and den Boer (2023) propose the Collude-or-Compete algorithm that solves the above two drawbacks under the multinomial logit demand. Assume that at each period, the consumer would buy the products from firm i with probability:

$$\lambda_i(\mathbf{p};\theta) := \frac{v_i(p_i;\theta)}{1+\sum_{i=1,-1} v_i(p_i;\theta)}, \tag{3.3}$$

.

where θ is a parameter unknown to the firms. The no-purchase probability is given by $\lambda_0(\mathbf{p};\theta) := \frac{1}{1+\sum_{i=1,-1} v_i(p_i;\theta)}$. A valuable property of the multinomial logit demand function is that when one firm observes a demand of 1, it can infer that the opponent has a demand of 0; conversely, when one firm observes a demand of 0, it can infer that the opponent has a demand of

either 0 or 1. The core of this collusive algorithm is that firms can reveal their demand through public prices, and the information of mutual demands enables the firms to consistently estimate θ, which provides the opportunity to sustain collusion.

Here I summarize several important notes on the Collude-or-Compete algorithm:

1. The algorithm consists of two modules: the collusive module and the competitive module. The algorithm starts with the collusive module, and will switch to the competitive module once it finds that the opponent doesn't use the collusive module.
2. In the competitive module, the algorithm employs a Kiefer-Wolfowitz recursion similar to that in Meylahn and V. den Boer (2022) to set prices, which allows it to learn the best response to the opponent's pricing. After a certain number of periods, the algorithm reverts to the collusive module.
3. During the collusive module, the algorithm utilizes publicly observable prices to disclose private information. Specifically, there are two types of private information:

   • First, the prices can reveal demand information. In the collusive module, it estimates θ using maximum-likelihood estimation by maximizing:

$$L_{t-1}(\theta) := \prod_{s\le t-1} \lambda_i(\mathbf{p}_s;\theta)^{d_{is}} \lambda_{-i}(\mathbf{p}_s;\theta)^{d_{-is}} \lambda_0(\mathbf{p}_s;\theta)^{1-d_{is}-d_{-is}} \tag{3.4}$$

   This estimation requires knowledge of the opponent's demand information d_{-is}. The method of revealing demand is as follows: each firm maintains three estimates for θ: $\hat{\theta}_{i,t}^{(0,0)}$, $\hat{\theta}_{i,t}^{(0,1)}$, and $\hat{\theta}_{i,t}^{(1,0)}$, which are updated using equation 3.4 under the assumption that the demands at time t are $(0,0)$, $(1,0)$, and $(0,1)$, respectively. The firm can then calculate the collusive prices $\mathbf{P}_t^{(0,0)}$, $\mathbf{P}_t^{(0,1)}$, $\mathbf{P}_t^{(1,0)}$ accordingly. It is important to note that if prices are public the demands are known to both firms and they start the collusive module simultaneously, $\hat{\theta}_{i,t}^{(\cdot,\cdot)}$ and $\hat{\theta}_{-i,t}^{(\cdot,\cdot)}$ should be identical at every period t, and the calculated collusive price pairs should also be identical.

   If firm i's demand at time t is 1, then at time $t+1$ the algorithm will set the price $\mathbf{P}_{it}^{(1,0)}$; conversely, if firm i's demand at time t is 0, although firm $-i$'s demand is unknown, the algorithm will set the price $\mathbf{P}_{it}^{(0,0)}$. However, at time $t+1$ the firm can infer that the demand of firm $-i$ at time t is 0 (1) if the opponet is posing price at $\mathbf{P}_{-it}^{(0,0)}$ ($\mathbf{P}_{-it}^{(0,1)}$). This 'reverse engineering' enables firms to communicate demand information solely through public price information.

   • Second, the price can reveal timing information to synchronize the switching of the two modules for the two firms. It can be observed that the collusive module can only learn the

collusive prices when both firms start using the collusive module simultaneously. In fact, during the first two time periods of the collusive module, the algorithm charges $p_{start,i}$ and $\kappa p_{start,i}$, where $0 < \kappa < 1$ is a fixed constant. Although κ is unknown to the opponent, this structured pricing data can help opponents learn the timing of the beginning of the collusive module.

4. The Collude-or-Compete algorithm is demonstrated to be resistant against other competitive pricing algorithms and is its own best response. Theoretical guarantees for convergence to supra-competitive prices are also provided, making it the first authentic collusive algorithm that does not rely on the demand being public.

The composite SPSA algorithm (Meylahn and V. den Boer (2022)) and the Collude-or-Compete algorithm (Loots and den Boer (2023)) are both authentic collusive algorithms with theoretical guarantees. Their presence indicates that algorithmic collusion may be more achievable than some regulators currently believe. The property that both algorithms need to know (or infer) the opponent's demand corresponds with Garcia, Janssen, and Shopova (2023), which suggests that the market price will increase in a duopoly pricing game when private stock information is shared. Currently, no authentic collusive algorithm has been developed without utilizing (including estimating) the opponent's demand information.

4. Regulate Algorithmic Collusion

4.1. Mutual Impact Between Algorithmic Collusion and the Market

Algorithmic collusion is detrimental to consumer welfare by inflating prices. However, research has revealed that the implications of algorithmic collusion for the market may extend beyond mere price increases. Yang, Lei, and Gao (2023) considers a duopoly market where each firm has a segment of *captive* customers who will only purchase from that firm and a segment of *contested* customers who may choose to buy from either firm. The firms can implement price discrimination across these different consumer segmentations and the regulator can enforce fairness regulations by requiring each firm to maintain a smaller price gap between the two market segments. A very interesting finding is that, numerically and analytically, the authors demonstrate that implementing fair pricing regulations can actually increase the likelihood of genuine collusion in some settings.

Several other studies have centered on the market's influence on the emergence of algorithmic collusion. Miklós-Thal and Tucker (2019) develop an analytical model in which the consumer's willingness to pay (demand) is unknown to the firms. It is found that more accurate demand forecasting algorithms can reduce potential opportunities for collusion. Johnson, Rhodes, and Wildenbeest (2023), Xu, Lee, and Tan (2023) investigate algorithmic collusion in online platforms where the platform's recommender system can segment the market into multiple sub-markets. In this market, sellers determine prices for various products through Q-learning. Not all products are accessible to every consumer; the platform recommends a bundle to consumers that includes a selection of products. Johnson et al. (2023) discovers that by restricting the size of the product assortment[22], the market price set by the firms tends to decrease (see Figure 12). Xu et al. (2023)'s numerical experiments reveal that the occurrence of algorithmic collusion via Q-learning depends on the platform's recommendation strategy: a profit-oriented recommender system tends to result in algorithmic collusion, whereas a demand-based system fosters competitive prices.

4.2. Regulatory Measures for Curbing Algorithmic Collusion

Among the existing research, three approaches have emerged for curbing algorithmic collusion: The first strategy entails the direct inspection of firms' pricing algorithms (Harrington (2018), Hartline, Long, and Zhang (2024)). The second approach involves intervention in the market, where authorities utilize reward-punishment mechanisms to deter collusion directly (Abada and Lambin (2023)). The third tactic focuses on designing market structures that inherently block algorithmic collusion (Johnson et al. (2023)).

Harrington (2018) is the first research addressing the regulation of algorithmic collusion, noting that currently there is no legal framework to regulate algorithmic collusion due to the difficulty in identifying the reward-punishment scheme. The author suggests that new laws should be enacted to inspect the internal algorithms used for pricing. For instance, regulators could input data similar to forced deviations in Calvano et al. (2019) into the algorithm, to determine if the algorithm generates prices with prohibited properties. This proposal may be too broad, and Hartline et al. (2024) further provides a more practical approach to audit.

[22] The recommender system is analogous to the assortment optimization problem for the platform, although neither paper explicitly uses this term.

Abada and Lambin (2023) introduces an alternative method. They study an electricity market where multiple firms sell or buy electricity units. The authors discovered that firms can rapidly learn to make seemingly collusive decisions under the Q-learning algorithm. Through numerical experiments, several methods are found to mitigate this collusive behavior:

1. Increasing the number of firms in the market can shift the game towards a competitive scenario. This aligns with prior findings of the *number effect* discussed in Subsection 2.3.
2. Regulators can actively intervene in collusion by acting as a firm within the market. Specifically, the regulator can implement a simple reward-punishment strategy, such as bidding aggressively when the market appears collusive and bidding conservatively when the market appears competitive. Numerical results indicate that this strategy can substantially enhance market welfare (see Figure 11).

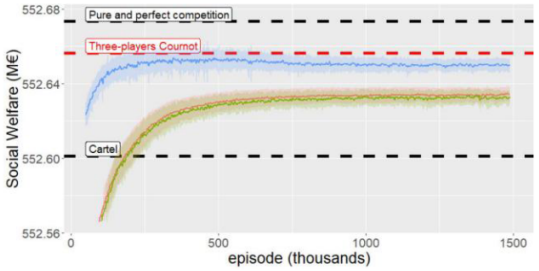

Figure 11
The figure depicts the dynamic of social welfare under different settings, where the blue line represents the regulator using the reward-punishment strategy and the green line represents the three-firm competing scenario. This is copied from figure 7 in Abada and Lambin (2023).

The authors also test alternative strategies. For example, they employ the same Q-learning algorithm, but instead of using the payoff as the reward, the regulator could use the instantaneous welfare[23] to update the Q-learning. However, experiments revealed that this approach has little impact on alleviating algorithmic collusion.

Johnson et al. (2023) centers on strategies to deter algorithmic collusion for platform owners. In this context, multiple firms employ Q-learning to set their prices, while the platform determines the assortment of products recommended to consumers. The platform's payoff includes both a commission fee derived from the total transactions and social welfare, indicating that the platform is interested in preventing widespread algorithmic collusion. The authors design some simple recommendation strategies which require little information to sustain the algorithmic collusion of the firms, the strategy contains two important features[24]:

[23] See equation B.1 in Abada and Lambin (2023)

[24] The first feature is derived from the PDP rule, while the second feature is derived from the DPDP rule, as detailed in Definitions 1 and 2 respectively from Johnson et al. (2023).

1. Limit the size of the assortment. The market will be more competitive when the number of recommended products decreases.
2. Recommend the lowest-priced product first, and if no other firm offers products with a significantly lower price, then never recommend the products of other firms.

Figure 12 presents the simulation results, with γ representing the parameter that controls the probability of recommending only one product; a lower γ value indicates a smaller assortment size. The red curve illustrates a stronger strategy, where the platform switches its recommendation set only when another firm significantly reduces the price beyond the current lowest price. It is evident that these measures are effective in mitigating algorithmic collusion.

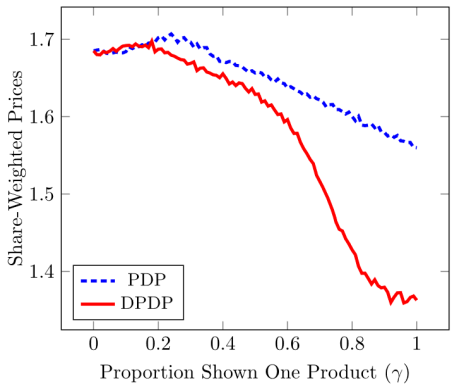

Figure 12
The figure depicts the market price given different γ and recommendation strategies. This is copied from figure 5 in Johnson et al. (2023).

5. Conclusion and Furthur Research Oppurtunities

In conclusion, the past and present research can be organized into the following insights:

• Many numerical studies have demonstrated that Q-learning and some other reinforcement learning algorithms can lead to supra-competitive prices in oligopoly pricing games.

• Some theoretical work proves that, under appropriate parameters, Q-learning (as well as some other algorithms) can converge to collusive prices in certain simple settings. Some of the convergences achieved are well-defined equilibriums, whereas the majority are not.

• Empirical evidence has identified that algorithmic pricing (not necessarily RL algorithms) can increase prices under competitive conditions;

• Although the current evidence is incomplete, and the seeming collusion is widely believed to be due to imperfect exploration, it still poses a great threat to social welfare because algorithmic collusion does not need to be a theoretically well-defined equilibrium; it just happens there.

• Collusions observed under RL algorithms can be categorized as spurious or genuine based on whether they exhibit a reward-punishment scheme for deviation. However, these types of algorithms can be outperformed by alternative algorithms, making them not likely to be practiced.

• Authentic collusion is a robust definition of algorithmic collusion, which must perform well against all alternative algorithms and be the best response toward itself (This has been beyond the scope of reinforcement learning).

• Authentic collusive algorithms have been developed, but current versions require to know (or infer) the opponent's demand information.

• Algorithmic collusion can be mitigated by the number effect. Regulatory intervention through active participation in the market, employing a reward-punishment strategy, can largely eliminate collusion.

• The design of online platforms can influence algorithmic collusion. Specifically, the platform's recommender system, which determines products that are visible to customers, can block or intensify algorithmic collusion.

Given the substantial threat and impact of algorithmic collusion on social welfare, it is vital to delve deeper into this topic. Future research could explore the following directions:

1. First, more numerical research could be conducted.

   • To date, the environmental settings in past numerical studies are quite simplified. Previous research usually assumed infinite capacity[25], no ordering cost, and model customer decision-making using a simple demand function. Future research could expand to more realistic settings, and consider inventory constraints and strategic consumer behaviors. Incorporating these constraints can make the simulations more representative of the reality.

   • Current numerical research in pricing games has primarily focused on 'static strategies', where the action space in the algorithms consists of deterministic prices. Future research could consider molecular strategies, such as *tit-for-tat* and *copycat* strategies similar to those considered in Kimbrough et al. (2005).

   Incorporating these 'dynamic strategies' into the action space allows the simulation outcomes to be more intuitively explained. Under the traditional action space, we can only determine whether an algorithm exhibits a reward-punishment scheme by inputting specific data and observing whether the output prices conform to certain patterns. However,

[25] Abada and Lambin (2023) is one exception.

this approach has defects; for example, Epivent and Lambin (2024) demonstrates that the reward-punishment scheme can respond not only to price decreases but also to price increases. In numerical experiments where the action space is composed of molecular strategies, the convergence of the Q-matrix can reveal the pricing strategy learned by the algorithms.

- A pressing concern arises within online platforms. Currently, many platforms, such as Airbnb and eBay, provide price recommendations to sellers operating on their platforms, with the final pricing decision left to the firms themselves. Hunold and Werner (2023) conducted laboratory experiments to demonstrate that when platforms use collusive strategies to recommend prices, the market price could be altered[26]. While platforms may not be so greedy in recommending prices through collusive strategies, it is natural for platforms to employ algorithms, such as Q-learning, to suggest prices. Whether Q-learning's recommendations can lead to collusive outcomes in the online market presents an intriguing avenue for further investigation.

- Another direction for numerical research is inspired by Spann et al. (2024). Beyond algorithmic collusion, can AI pricing algorithms lead to unfair pricing and facilitate automatic price discrimination in other market environments? This is a question that deserves further exploration.

- Furthur research could investigate whether AI algorithms would lead to algorithmic collusion in other disciplines, such as the financial markets (Cont and Xiong (2024), Dou, Goldstein, and Ji (2024)), auction games (Banchio and Skrzypacz (2022), Banchio and Mantegazza (2022)), assortment optimization (Aouad and den Boer (2021)), airline revenue management (Gu (2023)).

2. Second, there are several future directions for empirical research.

   - The number effect, verified by numerous numerical studies, suggests that as the number of competitors increases, the market price determined by Q-learning tends to decrease. When the number of firms reaches 10, the learned price converges to the static Nash Equilibrium price (Hettich (2021)). Wieting and Sapi (2021) provides some empirical evidence for this effect, but their works only reveal relation rather than causation. Future research could consider identifying the number effect.

[26] It should be noted that in Hunold and Werner (2023), contrary to intuition, the collusive pricing recommendation sometimes resulted in a lower market price compared to the baseline setting where the platform did not recommend prices.

- As discussed in Subsection 4.1, some analytical and numerical studies indicate that the market structure can influence the occurrence of algorithmic collusion. Future empirical research could focus on the mutual impact between the market structure and algorithmic collusion.

- Current empirical research has identified instances of algorithmic collusion, but the algorithms studied in these studies are not necessarily RL algorithms. Future research could leverage data with higher granularity to impact the impact of RL on market prices.

3. Thirdly, there is a need for more theoretical research.

   - We need a more rigorous framework to define genuine collusion, as illustrated in 3.1. The current framework does not provide a clear definition of the 'deviation' behavior, it just vaguely claims that deviation is a subset of alternative algorithms.

   Much of the analytical modeling research has focused on whether algorithmic collusion occurs by examining whether there exists a theoretical SPNE or MPE that supports the collusion (Yang et al. (2023), Miklós-Thal and Tucker (2019)). However, the absence of a collusive equilibrium does not suggest there is no potential for algorithmic collusion (see 2.6 for more details.). It's important to have a framework that unifies the gap between the observed collusion and theoretical collusion.

   - Future work could aim to provide more robust proofs of the convergence of Reinforcement Learning algorithms to collusive outcomes.

   - Meylahn and V. den Boer (2022) and Loots and den Boer (2023) present authentic collusive algorithms for duopoly markets; future research could explore the authentic collusive algorithms in oligopoly settings. Besides, current authentic algorithms need to infer the opponent's demand information; future research could design collusive algorithms without relying on this channel.

4. Research on the regulating aspect is lacking. Although in settings where a platform acts as an intermediary to determine which products are accessible to consumers, there are relatively practical solutions to deter algorithmic collusion (Johnson et al. (2023)); However, in settings where products are directly accessible to consumers without platform filtration, regulatory measures have a huge potential to be investigated (Harrington (2018), Abada and Lambin (2023), Hartline et al. (2024)).

   The development of more practical regulations against algorithmic collusion is imperative. Future research should investigate countermeasures against authentic collusive algorithms, including the composite SPSA (Meylahn and V. den Boer (2022)).

5. Finally, there is a huge potential for research in the field of platform operations management and information systems.

   • Johnson et al. (2023) and Xu et al. (2023) have examined the scenario where sellers employ Q-learning algorithms to price, and the platform determines which products to recommend to consumers. The recommender systems described in these papers may be more accurately modeled by assortment optimization problems in the OM literature. An emerging question is: under the current assortment optimization model, what will the market dynamics converge to when sellers use RL algorithms to price? Will the traditional assortment optimization scheme mitigate algorithmic collusion, or will it lead to 'meta-collusion'?

   Besides, the platform's payoff function may not solely be the commission rate from total transactions, but may also include a measure of social welfare[27]. Solving the optimal assortment under these constraints for the platform is an intriguing question for future research.

   Furthermore, platforms may not solely rely on assortment recommendation systems. In many cases, the platform has the ability to design coupons for different products. Future studies could devise the optimal coupon distribution strategy or explore alternative market structure modifications to improve platform payoffs and social welfare.

   • Information Systems (IS) study the interaction between IT artifacts and human beings and can be categorized into *behavioral science* and *design science*. As an IT artifact, AI pricing algorithms can be further investigated using IS methodologies from two perspectives: behavioral science to understand how algorithmic collusion shapes user behavior; and design science to devise a safe and practical pricing scheme for competitors.

References

Abada, I., & Lambin, X. (2023). Artificial intelligence: Can seemingly collusive outcomes be avoided? *Management Science*.

Adida, E., & Perakis, G. (2010). Dynamic pricing and inventory control: Uncertainty and competition. *Operations Research*, *58*(2), 289–302.

Aouad, A., & den Boer, A. V. (2021). Algorithmic collusion in assortment games. *Available at SSRN 3930364*.

Asker, J., Fershtman, C., & Pakes, A. (2022). Artificial intelligence, algorithm design, and pricing. In *AEA papers and proceedings* (Vol. 112, pp. 452–56).

[27] See Equation 3 in Johnson et al. (2023)

Asker, J., Fershtman, C., & Pakes, A. (2023). The impact of artificial intelligence design on pricing. *Journal of Economics & Management Strategy*.

Assad, S., Calvano, E., Calzolari, G., Clark, R., Denicolò, V., Ershov, D., ... others (2021). Autonomous algorithmic collusion: Economic research and policy implications. *Oxford Review of Economic Policy*, *37*(3), 459–478.

Banchio, M., & Mantegazza, G. (2022). Artificial intelligence and spontaneous collusion. *Available at SSRN*.

Banchio, M., & Skrzypacz, A. (2022). Artificial intelligence and auction design. In *Proceedings of the 23rd acm conference on economics and computation* (pp. 30–31).

BCG. (2020). *Debunking the myths of b2b dynamic pricing*. https://www.bcg.com/publications/2020/dynamic-pricing-b2b-myths/. ([Online; accessed June-2024])

BenMark, G., Klapdor, S., Kullmann, M., & Sundararajan, R. (2017). How retailers can drive profitable growth through dynamic pricing. *McKinsey. com*.

Besbes, O., & Sauré, D. (2016). Product assortment and price competition under multinomial logit demand. *Production and Operations Management*, *25*(1), 114–127.

Besbes, O., & Zeevi, A. (2015). On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, *61*(4), 723–739.

Brown, Z. Y., & MacKay, A. (2023). Competition in pricing algorithms. *American Economic Journal: Microeconomics*, *15*(2), 109–156.

Calder-Wang, S., & Kim, G. H. (2023). Coordinated vs efficient prices: The impact of algorithmic pricing on multifamily rental markets. *Available at SSRN 4403058*.

Calvano, E., Calzolari, G., Denicolò, V., Harrington Jr, J. E., & Pastorello, S. (2020). Protecting consumers from collusive prices due to AI. *Science (New York, N.Y.)*, *370*(6520), 1040–1042.

Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2019). Algorithmic pricing what implications for competition policy? *Review of industrial organization*, *55*, 155–171.

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

Calvano, E., Calzolari, G., Denicoló, V., & Pastorello, S. (2023). Algorithmic collusion: Genuine or spurious? *International Journal of Industrial Organization*, *90*, 102973.

Chen, M., & Chen, Z.-L. (2015). Recent developments in dynamic pricing research: Multiple products, competition, and limited demand information. *Production and Operations Management*, *24*(5), 704–731.

Chen, N., & Gallego, G. (2019). Welfare analysis of dynamic pricing. *Management Science*, *65*(1), 139–151.

Cho, I.-K., & Williams, N. (2024). Collusive outcomes without collusion: Algorithmic pricing in a duopoly model. *Available at SSRN 4753617*.

Clark, R., Assad, S., Ershov, D., & Xu, L. (2023). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy*.

Cont, R., & Xiong, W. (2024, April). Dynamics of market making algorithms in dealer markets: Learning and tacit collusion. *Mathematical Finance*, *34*(2), 467–521. doi: 10.1111/mafi.12401

Cooper, W. L., Homem-de-Mello, T., & Kleywegt, A. J. (2015). Learning and pricing with models that do not explicitly incorporate competition. *Operations research*, *63*(1), 86–103.

den Boer, A. V. (2023). Algorithmic collusion: A mathematical definition and research agenda for the OR/MS community. *Available at SSRN 4636488*.

den Boer, A. V., Meylahn, J. M., & Schinkel, M. P. (2022). Artificial collusion: Examining supracompetitive pricing by Q-learning algorithms. *Amsterdam Law School Research Paper*(2022-25).

den Boer, A. V. (2015). Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in operations research and management science*, *20*(1), 1–18.

Dolgopolov, A. (2024). Reinforcement learning in a prisoner's dilemma. *Games and Economic Behavior*, *144*, 84–103.

Dorner, F. E. (2021). Algorithmic collusion: A critical review. *arXiv preprint arXiv:2110.04740*.

Dou, W. W., Goldstein, I., & Ji, Y. (2024). Ai-powered trading, algorithmic collusion, and price efficiency. *Jacobs Levy Equity Management Center for Quantitative Financial Research Paper*.

Dutta, G., & Mitra, K. (2017). A literature review on dynamic pricing of electricity. *Journal of the Operational Research Society*, *68*, 1131–1145.

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

Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic collusion by large language models. *arXiv preprint arXiv:2404.00806*.

Fisher, M., Gallino, S., & Li, J. (2018). Competition-based dynamic pricing in online retailing: A methodology validated with field experiments. *Management science*, *64*(6), 2496–2514.

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

Gallego, G., & Hu, M. (2014). Dynamic pricing of perishable assets under competition. *Management Science*, *60*(5), 1241–1259.

Gallego, G., & Topaloglu, H. (2019). *Revenue Management and Pricing Analytics* (Vol. 279). New York, NY: Springer New York. doi: 10.1007/978-1-4939-9606-3

Garcia, D., Janssen, M. C., & Shopova, R. (2023). Dynamic pricing with uncertain capacities. *Management Science*, *69*(9), 5275–5297.

Gu, C. (2023). Can dynamic pricing algorithm facilitate tacit collusion? an experimental study using deep reinforcement learning in airline revenue management.

Hansen, K. T., Misra, K., & Pai, M. M. (2021). Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms. *Marketing Science*, *40*(1), 1–12.

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

Hartline, J. D., Long, S., & Zhang, C. (2024). Regulation of algorithmic collusion. In *Proceedings of the symposium on computer science and law* (pp. 98–108).

Hendon, E., Jacobsen, H. J., & Sloth, B. (1996). The one-shot-deviation principle for sequential rationality. *Games and Economic Behavior*, *12*(2), 274–282.

Hettich, M. (2021). Algorithmic collusion: Insights from deep learning. *Available at SSRN 3785966*.

Hu, J., & Wellman, M. P. (2003). Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, *4*(Nov), 1039–1069.

Hu, J., Wellman, M. P., et al. (1998). Multiagent reinforcement learning: theoretical framework and an algorithm. In *Icml* (Vol. 98, pp. 242–250).

Hunold, M., & Werner, T. (2023). Algorithmic price recommendations and collusion: Experimental evidence. *Available at SSRN*.

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

Kimbrough, S. O., Lu, M., & Murphy, F. (2005). Learning and tacit collusion by artificial agents in cournot duopoly games. *Formal modelling in electronic commerce*, 477–492.

Kimbrough, S. O., & Murphy, F. H. (2009). Learning to collude tacitly on production levels by oligopolistic agents. *Computational Economics*, *33*, 47–78.

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

Koirala, P., & Laine, F. (2024). Algorithmic collusion in a two-sided market: A rideshare example. *arXiv preprint arXiv:2405.02835*.

Lamba, R., & Zhuk, S. (2022). Pricing with algorithms. *arXiv preprint arXiv:2205.04661*.

Lei, Y., Jasin, S., & Sinha, A. (2018). Joint dynamic pricing and order fulfillment for e-commerce retailers. *Manufacturing & Service Operations Management*, *20*(2), 269–284.

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

Liu, J., Zhang, Y., Wang, X., Deng, Y., & Wu, X. (2019). Dynamic pricing on e-commerce platform with deep reinforcement learning: A field experiment. *arXiv preprint arXiv:1912.02572*.

Loots, T., & den Boer, A. V. (2023). Data-driven collusion and competition in a pricing duopoly with multinomial logit demand. *Production and Operations Management*, *32*(4), 1169–1186.

Martello, S. (2022). *Autonomous pricing using policy-gradient reinforcement learning* (Unpublished doctoral dissertation). Alma Mater Studiorum· University of Bologna.

Martínez-de-Albéniz, V., & Talluri, K. (2011). Dynamic price competition with fixed capacities. *Management Science*, *57*(6), 1078–1093.

Maskin, E., & Tirole, J. (1988). A theory of dynamic oligopoly, ii: Price competition, kinked demand curves, and edgeworth cycles. *Econometrica: Journal of the Econometric Society*, 571–599.

Melo, F. S. (2001). Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, 1–4.

Meylahn, J. M., & V. den Boer, A. (2022). Learning to collude in a pricing duopoly. *Manufacturing & Service Operations Management*, *24*(5), 2577–2594.

Miklós-Thal, J., & Tucker, C. (2019). Collusion by algorithm: Does better demand prediction facilitate coordination between sellers? *Management Science*, *65*(4), 1552–1561.

Misra, K., Schwartz, E. M., & Abernethy, J. (2019). Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, *38*(2), 226–252.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533.

Musolff, L. (2022). Algorithmic pricing facilitates tacit collusion: Evidence from e-commerce. In *Proceedings of the 23rd ACM conference on economics and computation* (pp. 32–33).

Özer, Ö., & Phillips, R. (2012). *The Oxford handbook of pricing management*. OUP Oxford.

Possnig, C. (2023). *Reinforcement learning and collusion*. Department of Economics, University of Waterloo.

Schlosser, R., & Boissier, M. (2018). Dynamic pricing under competition on online marketplaces: A data-driven approach. In *KDD* (pp. 705–714).

Schwalbe, U. (2018). Algorithms, machine learning, and collusion. *Journal of Competition Law & Economics*, *14*(4), 568–607.

Shi, Y., & Zhang, B. (2020). Multi-agent reinforcement learning in cournot games. In *2020 59th ieee conference on decision and control (cdc)* (pp. 3561–3566).

Siallagan, M., Deguchi, H., & Ichikawa, M. (2013). Aspiration-based learning in a cournot duopoly model. *Evolutionary and Institutional Economics Review*, *10*, 295–314.

Spann, M., Bertini, M., Koenigsberg, O., Zeithammer, R., Aparicio, D., Chen, Y., ... others (2024). *Algorithmic pricing: Implications for consumers, managers, and regulators* (Tech. Rep.). National Bureau of Economic Research.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

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

Wieting, M., & Sapi, G. (2021). Algorithms in the marketplace: An empirical analysis of automated pricing in e-commerce. *Available at SSRN 3945137*.

Williams, K. R. (2022). The welfare effects of dynamic pricing: Evidence from airline markets. *Econometrica*, *90*(2), 831–858.

Wittman, M. D., & Belobaba, P. P. (2019). Dynamic pricing mechanisms for the airline industry: A definitional framework. *Journal of Revenue and Pricing Management*, *18*, 100–106.

Xu, X., Lee, S., & Tan, Y. (2023). Algorithmic collusion or competition: the role of platforms' recommender systems. *arXiv preprint arXiv:2309.14548*.

Yang, Z., Lei, X., & Gao, P. (2023). Regulating discriminatory pricing in the presence of tacit collusion. *Available at SSRN*.