Algorithmic and Human Collusion

Algorithmic and Human Collusion

Tobias Werner[1,*]

Abstract: I study self-learning pricing algorithms and show that they are collusive in market simulations. To derive a counterfactual that resembles traditional tacit collusion, I conduct market experiments with humans in the same environment. Across different treatments, I vary the market size and the number of firms that use a pricing algorithm. I demonstrate that oligopoly markets can become more collusive if algorithms make pricing decisions instead of humans. In two-firm markets, prices are weakly increasing in the number of algorithms in the market. In three-firm markets, algorithms weaken competition if most firms use an algorithm and human sellers are inexperienced.

Keywords: Artificial Intelligence, Collusion, Experiment, Human–Machine Interaction

Classification: C90, D83, L13, L41

[*]**Correspondence address:** tobias.felix.werner@soton.ac.uk; Draft compiled on December 23, 2025

1 Introduction

The use of autonomous pricing algorithms is on the rise in various industries.[1] When firms use these tools, the pricing decision for a given product is outsourced from the human decision-maker to a computer algorithm. While in the past, most pricing algorithms have been rule-based with rules defined by the seller, there is a recent evolution towards self-learning algorithms (Ezrachi and Stucke, 2017). These self-learning algorithms develop strategies to achieve a specific goal, such as maximising the firms' profits, without explicit instructions.

There are concerns among competition authorities (e.g., Bundeskartellamt and Autorité de la concurrence, 2019; Competition & Markets Authority, 2021) and academic scholars (e.g., Ezrachi and Stucke, 2016, 2017; Mehra, 2016) that pricing algorithms could not necessarily learn to price products more efficiently, but also that there exists a possibility that they learn to collude tacitly.[2] In other words, algorithms could learn that collusion benefits the firm without explicit instructions or communication with other algorithms.

While recent papers by Calvano *et al.* (2020b) and Klein (2021) show that Q-learning algorithms, a popular type of self-learning algorithm, can learn to be collusive, it remains unclear whether pricing algorithms are more collusive than humans and, therefore, harm competition. Tacit collusion in traditional markets amongst human decision-makers is a well-documented phenomenon in empirical and experimental economics.[3] To assess the (anti-)competitive effects of algorithms, it is necessary to establish a suitable baseline. Accordingly, this paper

[1]The European Commission (2017) finds that two-thirds of sellers in digital markets use pricing tools. Prominent examples are Amazon (Chen *et al.*, 2016b; Musolff, 2022) or the petrol market (Assad *et al.*, 2024).
[2]For recent discussions about the possible policy and legal implications of algorithmic pricing, see Kühn and Tadelis (2017); Schwalbe (2018); Calvano *et al.* (2019, 2020a) and Assad *et al.* (2021); Harrington (2018).
[3]See e.g., Byrne and De Roos (2019); Miller and Weinberg (2017); Borenstein and Shepard (1996); Davies *et al.* (2011) for empirical, and Engel (2015); Horstmann *et al.* (2018) for experimental evidence.

addresses the research question: Are Q-learning-based pricing algorithms inherently more collusive than their human counterparts?

Furthermore, many modern markets do not consist solely of algorithms or humans. Often, both can interact with each other in the same market environment. For instance, different firms may have different levels of sophistication or financial constraints when deciding on pricing tools. According to Chen *et al.* (2016b), only one-third of vendors selling the most popular products on amazon.com use some form of pricing algorithm, which gives rise to a mixture in market composition. Algorithms that are collusive amongst themselves may behave differently when interacting with humans. Algorithmic collusion in isolation does not necessarily suggest that they behave anti-competitive in these "mixed" markets. Therefore, a crucial aspect to investigate is: What market outcomes should be expected if self-learnt pricing algorithms and humans interact in the same environment?

This paper provides a counterfactual for algorithmic collusion for a wide range of possible market compositions and highlights the impact of self-learning algorithms on competition. To examine whether commonly used Q-learning algorithms make markets more collusive relative to the status quo of human collusion, I apply a two-step approach. In the first step, I consider self-learning pricing algorithms in an extensive simulation study. I test whether algorithms learn to set supracompetitive prices and suitable strategies to support these prices as a collusive outcome. Here, I closely follow the approach from Calvano *et al.* (2020b) by focusing on Q-learning algorithms, but I consider a different, more tractable market environment. In the second step, I conduct market experiments where humans compete against each other or with self-learnt pricing algorithms. In the experiments, I closely mimic the market environment from the simulations. Across different treatments, I

vary the market composition between algorithms and humans and the number of firms in the market. The experimental approach allows me to consider tacit collusion in a controlled setup and study the underlying mechanics. My design enables me to observe humans and algorithms in the same environment and, thus, to analyse whether Q-learning algorithms promote collusion.

I find evidence that Q-learning algorithms foster tacit collusion in duopolies. In duopolies, where humans and algorithms compete, prices are similar to markets with only humans. Two-firm markets with only algorithms are always more collusive than markets with humans. Hence, market prices are (weakly) increasing in the number of algorithms in duopolies and can foster collusion if all firms use one. In triopolies, there is a non-linear relationship between the number of firms with a pricing algorithm and the level of tacit collusion. Markets where a single firm utilises a pricing algorithm are more competitive than markets with only humans. Yet, as more firms use pricing algorithms, market prices can increase and may even exceed prices in human markets. This effect is especially noticeable when humans lack experience.

Similar to Calvano *et al.* (2020b) and Klein (2021), Q-learning algorithms learn to punish price deviations. As I consider a stylised market environment, I can interpret the strategies of the algorithms. The most successful algorithms learn a win-stay lose-shift strategy that is popular for the iterated prisoner's dilemma. The outcomes in mixed markets vary greatly as humans choose heterogeneous strategies when playing against the algorithms.

While there exists reoccurring support for the hypothesis that self-learning algorithms can learn to set non-competitive prices and develop reward-punishment strategies (Klein, 2021; Calvano *et al.*, 2020b, 2021; Abada and Lambin, 2023; Johnson *et al.*, 2023; Asker

et al., 2024), it is unclear how algorithmic collusion compares to human collusion.[4] Market environments in previous studies on algorithmic collusion deviate substantially from the setting used in experimental market games. My design allows me to compare the outcomes of pricing algorithms to human pricing directly, as I can observe both in the identical market environment. A recent paper by Assad *et al.* (2024) identifies the adoption of algorithms in the German gasoline market. Within duopoly and triopoly markets, price margins rise as both firms in the market begin to utilise an algorithm. The effect is comparable to my findings for two-firm markets. For the market studied by Assad *et al.* (2024), the exact algorithms are unobservable as they are usually proprietary. The combination of simulations and laboratory experiments enables me to examine human and algorithmic strategies to study the underlying mechanics that drive these effects.

This research also allows studying cooperation between humans and algorithms, contributing to a growing literature in computer science and experimental economics. In computer science, the design of cooperative algorithms in repeated games is an active research area (e.g., Crandall *et al.*, 2018; Lerer and Peysakhovich, 2017). Here, cooperative algorithms are often the explicit objective. I consider a popular self-learning algorithm that can be attractive as a pricing tool. While cooperation might be an outcome, it is not the initial design objective.

In experimental economics, the focus is often on deterministic algorithms that do not learn the strategy themselves but instead follow pre-defined rules by the experimenter (see March, 2021, for a recent literature review). For example, Huck *et al.* (1999) and Huck

[4]Besides self-learnt collusion, algorithms could also influence competition by offering better demand predictions (Miklós-Thal and Tucker, 2019; O'Connor and Wilson, 2021) or by serving as commitment devices (Brown and MacKay, 2023; Leisten, 2024). Furthermore, Harrington (2022) argues that outsourcing the development of algorithms to a third-party developer can affect market outcomes. For a general discussion of the possible effects algorithms can have on the economy, see Agrawal *et al.* (2019).

et al. (2004a) study simple learning processes inspired by human behaviour in experimental games, while Duersch *et al.* (2010) examine how humans interact with these rule-based algorithms. There are parallels to this paper, as the algorithms used here also employ a fixed strategy when interacting with humans and do not learn anymore during the experiments. However, unlike previous studies, the algorithms learn these strategies autonomously before the experiments, and the experimenter does not design them. Thus, while the algorithm operates deterministically during the experiment, it differs from purely rule-based algorithms as it learn the strategy by itself in a training phase.

Normann and Sternberg (2023) study collusion in mixed markets with humans and algorithms. They consider a tit-for-tat algorithm and find that in three-firm markets where one of the three firms uses a tit-for-tat algorithm, outcomes are more collusive than in markets where all firms are human. The authors vary whether participants know if they play against a computer or a person, and find no differences in this domain. My approach differs as I study self-learning algorithms and the strategies which they develop by themselves. It allows analysing the entire array of market composition, as I can observe algorithmic and human markets as well as mixed markets. Hence, I can directly compare algorithmic and human collusion and investigate the effect of Q-learning-based pricing algorithms on a wide range of scenarios.

The remainder of the paper is structured as follows. In Section 2, I introduce the market environment I use in all simulations and experiments with humans. Then, in Section 3, I explain the main concepts of the Q-learning algorithm, which I consider in this study, and discuss the motivation for focusing on this particular type of algorithm. After discussing the

experimental design in Section 4 and the hypotheses in Section 5, I present the results in Section 6. Section 7 discusses the implications of my findings and concludes.

2 Market environment

I consider a Bertrand market environment, which is commonly used in the experimental economics literature on collusion (see, for instance, Fonseca and Normann, 2012; Horstmann *et al.*, 2018). There are $N \in \{2, 3\}$ firms in the market, which face a perfectly inelastic demand function and have zero marginal costs. Each firm produces the same homogeneous good. The market consists of $m = 60$ computerised consumers, who are all willing to purchase exactly one unit of this good in each round and have a maximum willingness to pay of $\overline{p} = 4$. The price of firm i in period t is denoted by $p_t^i \in \mathscr{P} := \{0, 1, 2, ..., 5\}$. Consumers buy the good at the lowest offered price. The market is shared equally if multiple firms offer the lowest price in a given round.

Firms are always either represented by a human or by a Q-learning algorithm. This market environment is the same for the simulation study and all experimental treatments. It allows me to directly compare the simulation and outcomes and derive a counterfactual for algorithmic collusion. Firms compete in an infinitely repeated game with a discount rate of $\delta = 0.95$.

To mimic the features of an infinitely repeated game in the experimental treatments, I use a repeated game with a random stopping rule, where the continuation probability for playing another round is given by 95%. Using smaller discount factors, this method of translating

infinitely repeated games into indefinitely repeated games was first introduced by Roth and Murnighan (1978).

There exists a stage game Nash equilibrium at $p^{NE} = 1$. The monopoly price of $p^M = \overline{p} = 4$ maximises joint profits.[5] When all firms charge the same price, the profit is given by $\pi_t^i = p\,m/N$. The profit for a single deviating firm is $\pi_t^i = (p-1)\,m$. Collusion is sustainable at the monopoly price for the given discount factor, for instance, by grim trigger strategies.

While this environment is less complex than many actual markets, it yields a suitable setting for my design as it distils the main components of price competition when studying collusion.[6] It is arguably easy to understand for experimental subjects due to its simple mechanics. Crucially, the environment is tractable, which allows for the analysis of the strategies algorithms and humans learn in the game. Also, the environment offers a different extension to algorithmic price competition to markets with a perfectly inelastic demand function, which has not been studied before.

3 Pricing algorithms

Following the approach independently proposed by Calvano *et al.* (2020b) and Klein (2021), I utilise Q-learning algorithms (Watkins, 1989; Watkins and Dayan, 1992) to study the collusive effects of self-learning pricing algorithms.[7] I first discuss the rationale for focusing

[5]There are two prices ($p = 5$ and $p = 0$) which are (weakly) dominated. I include both in the set of possible prices $\mathscr{P}$ to rule out that convergence to the boundaries of the price set is equivalent to collusion, or competition at the $p = 1$ Nash equilibrium. The full set of symmetric pure-strategy Nash equilibria is $p \in \{0, 1, 2\}$ for $N = 2$ and $p \in \{0, 1\}$ for $N = 3$ for the stage game. I focus on $p^{NE} = 1$ as the competitive benchmark, as $p = 0$ is payoff dominated and $p = 2$ is not an equilibrium for $N = 3$.

[6]The environment shares strategic similarities with the prisoner's dilemma game but includes more actions to capture pricing dynamics observed in markets. For a literature review on the behaviour of humans in the prisoner's dilemma, see Mengel (2018) and Dal Bó and Fréchette (2018).

[7]Earlier work by Waltman and Kaymak (2008) shows that Q-learning algorithms can converge to non-competitive quantities in a Cournot framework. However, they do not obtain collusion as algorithms also

on the Q-learning algorithm. Then, I introduce Q-learning and the simulation setup used in this study.

3.1 Background and motivation

There are several reasons for focusing on Q-learning when studying the ability of self-learning pricing algorithms to collude.[8] Q-learning is one of the most relevant and popular reinforcement learning algorithms, and many of the most successful state-of-the-art reinforcement learning algorithms build on the main ideas of Q-learning (see, for instance, Arulkumaran *et al.*, 2017, for an overview).

Q-learning is designed to solve Markov decision processes with an ex-ante unknown environment. In other words, Q-learning algorithms must learn about the environment by themselves and are not instructed to follow a particular strategy. The algorithms usually learn their strategies by competing against other independent algorithms, that strive themselves to develop optimal strategies. Therefore, the algorithms must continuously adapt during learning as the competitors also regularly change their strategies. Arguably, this fosters the learning of strategies that extrapolate to different competitors. The learning approach often results in superior performance from reinforcement learning algorithms compared to human strategies or if they would learn only against humans.[9] This potential for

learn this behaviour if they are memoryless. Hence, punishment strategies, which are essential for collusion to be sustainable in the long run, could never arise.

[8]While most studies on algorithmic collusion concentrate on Q-learning, there are notable exceptions. For instance, Hansen *et al.* (2020) examine pricing algorithms using an Upper Confidence Bound algorithm in a multi-armed bandit framework, where supracompetitive prices emerge from correlated experimentation. Additionally, Hettich (2021) and Jeschonneck (2021) explore reinforcement learning methods that rely on function approximation, which may result in different pricing behaviours. For a recent discussion on the limits of Q-learning in studying algorithmic collusion, see den Boer *et al.* (2024).

[9]Reinforcement learning techniques have been shown to outperform humans and other algorithms in a variety of applications, including Atari video games (Mnih *et al.*, 2015), the board game Go (Silver *et al.*, 2016), chess (Silver *et al.*, 2018), ridesharing optimisation (Qin *et al.*, 2022), or inventory management in e-commerce (Madeka *et al.*, 2022).

superior performance provides an apparent reason for firms to employ such algorithms in pricing.

There is also direct evidence that reinforcement learning algorithms like Q-learning are used in real-world pricing systems. For instance, researchers from Alibaba, one of China's largest retailers, report on a pricing algorithm that uses a form of Q-learning algorithms, which outperforms previous pricing methods (Liu *et al.*, 2019). Similarly, Chen *et al.* (2023) describe a price discount algorithm used by the leading budget hotel group in China that utilises reinforcement learning. Moreover, Calzolari and Hanspach (2024) note that 27% of third-party pricing tools, that sellers on platforms like amazon.com use, claim to utilise self-learning algorithms. It highlights that self-learning algorithms are popular in practise and that even smaller, possibly less sophisticated, firms have access to this pricing technology.

Furthermore, compared to other reinforcement learning algorithms, Q-learning remains interpretable. Q-learning algorithms generate strategies that are essentially rules that are directly observable and map past market outcomes to future prices. However, unlike fixed rules chosen by a human, the algorithm learns these rules by themselves.

Despite the popularity of reinforcement and Q-learning, it is crucial to note that other forms of algorithmic pricing exist. Large e-commerce companies like Zalando, Zara, Asos, and Walmart employ pricing algorithms that combine machine learning with other numerical optimisation techniques (Kunz *et al.*, 2023; Li *et al.*, 2021a; Loh *et al.*, 2022; Mehrotra *et al.*, 2020; Caro and Gallien, 2012). Amazon describes its own pricing algorithms as using machine learning without specifying the exact approach (Cooprider and Nassiri, 2023). Assad *et al.* (2024) and Derakhshan *et al.* (2016) report on neural network-based gasoline pricing algorithms. While these algorithms are similar to reinforcement learning in the sense that

they make autonomous decisions, they rely on predefined data sets rather than learning by themselves from interacting with an environment.

Additionally, rule-based algorithms are still employed, where pricing managers set rules on how prices should change based on changes in the market. Musolff (2022) and Wang *et al.* (2023) discuss how third-party sellers use such tools on platforms like amazon.com, allowing vendors to maintain some control over their pricing strategies. Calzolari and Hanspach (2024) elicit that while 56 % of repricer providers in their sample claim to use some form of AI, 44 % use less sophisticated methods like fixed rules.

Importantly, while this paper focuses on Q-learning algorithms when investigating human and algorithmic collusion, given its relevance in academic research and real-world applications, it is worthwhile for future research to explore other forms of algorithms that are prevalent in practise, such as rule-based ones or those that use other forms of machine learning. For the sake of clarity, in the remainder of this paper, whenever I refer to "algorithms", I specifically mean Q-learning algorithms. In the following, I discuss some of the general concepts in Q-learning.

3.2 Q-learning: Optimisation problem and learning

In each period t, a Q-learning algorithm, often called an agent, observes the current state $\mathbf{s}_t \in \mathbf{S}$ of its environment and chooses some action $a_t \in \mathbf{A}$. Here, $\mathbf{A}$ is the set of feasible actions, and $\mathbf{S}$ is the set of possible states. Picking the action results in a reward signal $\pi_t \in \mathbb{R}$ and the next state $\mathbf{s}_{t+1} \in \mathbf{S}$. The agent's objective is to maximise the sum of discounted future expected rewards given the current state $\mathbf{s}_t$ over $\mathbf{A}$. The Bellman equation commonly

expresses this maximisation problem

$$V(\mathbf{s}_t) = \max_{a_t}\{\mathbb{E}[\pi_t|\mathbf{s}_t, a_t] + \delta\mathbb{E}[V(\mathbf{s}_{t+1})|\mathbf{s}_t, a_t]\} \tag{1}$$

with $\delta \in [0, 1)$ being the discount rate. The Bellman equation described by Equation 1 is recursive. The value of being in state $\mathbf{s}_t$ is given by the current reward signal π_t plus the discounted value of the continuation state $\mathbf{s}_{t+1}$.

In Q-learning, the environment is typically unknown to the agent. Before learning, the agent does not know which actions result in which states or which state-action combinations lead to which rewards. Furthermore, the environment might be non-stationary in the sense that the same state-action combinations in period t may lead to a different reward and another continuation state in a different period t'.

In Q-learning, the Bellman equation is rewritten as the Q-function

$$Q(\mathbf{s}_t, a_t) = \mathbb{E}[\pi_t|\mathbf{s}_t, a_t] + \delta\mathbb{E}[\max_{a} Q(\mathbf{s}_{t+1}, a)|\mathbf{s}_t, a_t] \tag{2}$$

In this paper, $\mathbf{A}$ and $\mathbf{S}$ are finite, and hence the Q-function is given by a $|\mathbf{S}| \times |\mathbf{A}|$ matrix, where $Q(\mathbf{s}_t, a_t)$ represents the expected net present value of picking action a_t in state $\mathbf{s}_t$. The Q-learning agent repeatedly interacts with the environment to iteratively update the cells of its Q-matrix to obtain an approximation for the state-action value $Q(\mathbf{s}_t, a_t)$ for each state-action combination. By doing so, the agent aims to maximise its expected long-run reward.

In all simulations, the Q-matrix is initialised with random numbers drawn from a uniform distribution with support on the unit interval.[10] For each subsequent iteration t, the agent picks some action a_t conditional on the current state $\mathbf{s}_t$, which yields π_t and $\mathbf{s}_{t+1}$. Then, the Q-matrix gets updated as the weighted average of the past estimate of $Q(\mathbf{s}_t, a_t)$ and the newly learnt value

$$\mathbf{Q}_{t+1}(\mathbf{s}_t, a_t) = (1-\alpha)\mathbf{Q}_t(\mathbf{s}_t, a_t) + \alpha(\pi_t + \delta \max_a \mathbf{Q}_t(\mathbf{s}_{t+1}, a)) \tag{3}$$

where $\alpha \in (0, 1)$ is referred to as the *learning rate*. For the subsequent states, the same procedure is applied in each iteration until some convergence rule is met.

In each round, the agent picks the current optimal action a_t^* with probability $(1 - \varepsilon_t)$ with $\varepsilon_t \in [0, 1]$. With probability ε_t the algorithm selects a random action from $\mathbf{A}$. It balances the trade-off between exploring the environment and exploiting the knowledge the agent already has of the profitability of certain state-action combinations (see, for instance, Klein, 2021, for further discussions on this). The exploration probability decays over time $\varepsilon_t = e^{-\beta t}$ for some small $\beta > 0$. Like this, the agent explores more at the beginning of the learning process when Q-matrix is rather uninformative but chooses the action, which offers the highest expected long-run reward, more often in later periods. Eventually, the agent no longer explores but always picks the action with the highest expected value given that $\lim_{t\to\infty} \varepsilon_t = 0$.

[10] Klein (2021) and Abada and Lambin (2023) initialise the Q-matrix with zeros. Calvano *et al.* (2020b) and Johnson *et al.* (2023) use an initialisation that corresponds to the discounted profit if all firms randomise their prices. Calvano *et al.* (2020b) show that outcomes are similar for different initialisation of the Q-matrix. In Appendix A.3, I consider an initialisation of the Q-matrix that uses human data from the experimental treatments. Also here the results remain similar.

3.3 Simulation setup

In each simulation, the agent plays against other agents in the same market. The agents share the same α and β but are otherwise disconnected. Each agent picks their action independently, and each agent's learning process is separated from the others. The Q-matrix is initialised independently for each agent and updated only based on their own reward. The agents learn by playing against each other simultaneously in the same environment. That is, the agents learn "online" by interacting with the environment instead of "offline", where the agents would learn from a static data set (see Agarwal *et al.*, 2020, for details on the two learning approaches). This form of decentralised learning is a common approach in multi-agent reinforcement learning applications (see, for instance, Littman, 1994; Busoniu *et al.*, 2008).

To determine the state of convergence, I follow the approach by Calvano *et al.* (2020b) and look at the cells of the Q-matrix. If the best action for each state does not change for 100,000 subsequent periods, I assume that the agents in the market have converged and found a stable strategy. I refer to the strategy the Q-learning algorithms learnt at convergence as their limit strategy.

The action a_t of the agent corresponds to a price, and the set of possible actions corresponds to the price set, while the economic profit obtained in each period is the reward signal for the Q-learning algorithms. Similar to Calvano *et al.* (2020b) and Johnson *et al.* (2023), I define the state of the environment for each agent as the set of past prices from the previous period $\mathbf{s}_t = \{p_1^{t-1}, ..., p_N^{t-1}\}$. Notably, this state representation corresponds to memory-one strategies, which humans predominantly use in the prisoner's dilemma and market games (Dal Bó and Fréchette, 2019; Romero and Rosokha, 2018; Wright, 2013). Calvano *et al.*

(2020b) and Kasberger *et al.* (2023) also consider state representations that allow for a two-period and three-period memory. This larger memory does not improve the performance of the algorithms. Importantly, it is straightforward to construct memory-one strategies that make collusion incentive-compatible in two and three-firm markets.[11]

The learning parameter α and the exploration decay parameter β are not learnt by the algorithm, and a wide range of possible parameterisations and, thus, market outcomes are possible. Furthermore, the learning process of the algorithms is partially stochastic due to the exploration mechanism, which introduces randomness into the decision-making process. I consider a parameter grid with $\alpha \in [0.025, 0.25]$ and $\beta \in [1 \times 10^{-8}, 2 \times 10^{-5}]$ with 100 points in each dimension evenly spaced from one another. For each grid point, I simulate 1,000 distinct markets that differ in the underlying stochastic process that governs exploration. Hence, I run a total of 10,000,000 simulations for each market size. The grid and the simulation setup, again, follows Calvano *et al.* (2020b) and Johnson *et al.* (2023).

To implement the simulations, I use Python with the packages "numpy" and "numba" for efficient scientific computing. The simulations were executed on the high-performance cluster of the University of Duesseldorf. While each simulation runs within seconds, the extensive number of simulations made the cluster a more practical solution.

3.4 Performance evaluation

Profitability and prices I focus on the (average) market price upon convergence to measure tacit collusion. Furthermore, given the definition of the Bellman equation, $V(\mathbf{s})$

[11] For instance, consider a memory-one strategy that mimics a grim trigger strategy in the sense that the agent always plays the monopoly price in the state $\mathbf{s} = (p^M, p^M, p^M)$ but chooses the $p = 1$ Nash equilibrium in any other state. This strategy is a possible outcome for the simulations from an ex-ante perspective and makes collusion sustainable for the given discount factor of $\delta = 0.95$.

provides an unbiased estimate of the expected sum of future discounted rewards for a given state $\mathbf{s}$. Thus, $V(\mathbf{s})$ is a natural measure for the profitability of agent i in a given state $\mathbf{s}$. I utilise this direct interpretability and use $V(\mathbf{s})$ as an additional profitability measure.

Optimality High profitability alone does not necessarily imply that the agent has learnt an optimal strategy against its competitors. To derive a measure of how well the agent does in comparison to how well it could do, I derive the optimality of the agent for state $\mathbf{s}$ as

$$\Gamma_i(s) = \mathbb{1}\{\arg\max_a \mathbf{Q}_i(\mathbf{s}, a) = \arg\max_a \mathbf{Q}_i^*(\mathbf{s}, a)\} \tag{4}$$

where $\mathbf{Q}_i^*(\mathbf{s}, a)$ is the optimal Q-matrix, assuming that the opponents play according to their limit strategies. Here, $\mathbb{1}$ denotes the indicator function.

I can obtain this optimal Q-matrix in the following way: After convergence, the strategies of the other agents, which learnt in the same environment, are held constant. Thereby, the environment becomes stationary as the state-transition probabilities do not change anymore. Next, I initialise a new agent, which competes against the limit strategy of the opponents. I utilise dynamic programming to repeatedly iterate over each state-action combination of its Q-matrix until convergence. Within this stationary environment, convergence is guaranteed. The Q-matrix of the new agent corresponds to the optimal Q-matrix $\mathbf{Q}_i^*(\mathbf{s}, a)$ that the actual agent could have learnt against the limit strategy of the other agents.

Note that $\Gamma_i(\mathbf{s}) = 1$ implies that the agent has learnt to play a Nash equilibrium for state $\mathbf{s}$. The agent has learnt a subgame perfect Nash equilibrium if and only if $\overline{\Gamma} = \frac{1}{|\mathbf{S}|}\sum_{\mathbf{s}} \Gamma_i(\mathbf{s}) = 1$. For $\overline{\Gamma} < 1$ there are states for which the agent did not learn to play a Nash equilibrium.

3.5 Main parameterisation of the algorithm

I study experimental treatments in which humans compete against algorithms in the same market. To conduct laboratory experiments in these "mixed" markets, I have to decide on a specific parameterisation of the agent with a specific stochastic process. To select a specific algorithm for this purpose, I take the perspective of a firm that would want to deploy a pricing algorithm to a market. This firm would want the pricing algorithm to meet three criteria: (i) it should be profitable, (ii) it should be optimal in the sense that it is not easily exploitable by other market participants, and (iii) it should allow for forgiveness in the event of deviations.

Considering these objectives in isolation is insufficient when selecting an algorithm to deploy to the market. High profitability does not necessarily imply high optimality or forgiveness, and vice versa. For instance, it is vital to rule out that a lack of sophistication in algorithms drives high profitability.[12] This can happen if the agents in the environment jointly converge to a seemingly collusive outcome in which they price at the monopoly price but fail to learn a strategy that accounts for certain deviations. Such a myopic strategy is unlikely to perform well against new competitors, resulting in lower profitability than in the simulation. Similarly, forgiveness is crucial as it ensures the algorithm can recover from deviations, which are common in real-world markets due to mistakes or experimentation.[13] Otherwise, after an unintentional deviation, the algorithm would be stuck in low-profit states without the ability to return to a collusive outcome.

[12] The situation can arise, for example, if the agent's exploration is limited.

[13] Too frequent deviations are common in strategic environments, as highlighted by Duffy *et al.* (2024). They show in the indefinitely repeated prisoner's dilemma that participants deviate too frequently even if they know they are playing against a grim trigger strategy and it is a subgame perfect equilibrium to cooperate. Such "mistakes" would likely be even more prevalent in a pricing game that is more complex and where participants do not know the competitor's strategy.

When all three objectives – profitability, optimality, and forgiveness – are ensured jointly, the agent has learnt a strategy that can generate profits but also takes into account the strategic element of the environment and is capable of returning to collusive outcomes after deviations. This combination increases the likelihood that the algorithms perform well against new (possibly human) competitors.

I propose the following criterion for agent i that combines all three performance measures:

$$\Psi_i = \frac{\overline{\Gamma}_i}{|\mathbf{S}|} \sum_{\mathbf{s}} V_i(\mathbf{s}). \tag{5}$$

Since $\overline{\Gamma}_i \in [0, 1]$, the selection criterion is the average profitability over the entire state space shrunk towards zero by the degree of suboptimality. Hence, $\overline{\Gamma}_i$ can be interpreted as a shrinkage penalty in this context. The intuition is that high profitability is only valuable if other players cannot exploit the agent easily with a possibly more sophisticated strategy.

Focusing on profitability across all states, rather than just the state of convergence, helps select forgiving limit strategies that can recover from deviations. Different states are potentially relevant in the experiments with humans, as their strategies can differ from algorithmic ones. An algorithm that performs well only on the equilibrium path, like grim trigger, would struggle to return to collusion after a disruption, leading to sustained low profits after an unintentional deviation. By evaluating profitability across all states, the selection criterion favours algorithms that are potentially profitable in a wide range of scenarios by having the capability of returning to a collusive state even after a deviation.[14]

[14]While considering the entire state space when selecting an algorithm is reasonable, the selection criterion $\overline{\Gamma}_i V_i(\mathbf{s}^*)$, where $\mathbf{s}^*$ is the state of convergence, leads to similar strategies being selected in the simulations.

For treatments in which algorithms interact with humans, I select the algorithm from the simulation in which $\overline{\Psi} = \frac{1}{N}\sum_i^N \Psi_i$ is maximised over the parameter grid for α and β. I will refer to it as the selected algorithm. Also, for markets with only algorithms, I will focus on the selected algorithm's parameterisation but report the results for other parameterisations in the Appendix A.2.

4 Experimental design

4.1 Treatments

I consider five experimental treatments and two treatments based on simulations. Across treatments, I vary the market composition between algorithms (A) and humans (H), and the number of firms in the market (see Table 1). I label the treatments with the number of human firms followed by the number of firms that use an algorithm. For example, treatment 2H1A stands for two human players and one algorithmic player operating in one market.

Table 1: Treatment composition

Number of Human Players	Number of Algorithms			
	0	1	2	3
3	3H0A	-	-	-
2	2H0A	2H1A	-	-
1	-	1H1A	1H2A	-
0	-	-	0H2A	0H3A

Thus, I consider treatments without any algorithms (2H0A and 3H0A) and without any humans (0H2A and 0H3A). Comparisons between these treatments reveal whether algorithms are more collusive than humans. Additionally, I consider treatments in which

humans compete against algorithms (1H1A, 1H2A and 2H1A). I utilise these treatments to examine the way humans and pricing algorithms interact with each other. Furthermore, they show if increasing the share of algorithms in the market fosters tacit collusion for different market sizes.

4.2 Procedure

General procedure for all experimental treatments Each experimental treatment is repeated for three supergames to observe learning effects.[15] Within each supergame, there is a fixed group composition. Across supergames, I use a perfect stranger matching scheme. This matching scheme is common knowledge. Hence, participants know they interact with each person only within one supergame throughout the entire experiment. It rules out any reputation effects that could be present otherwise.

In my experiment, each round has a continuation probability of 95% in each supergame. Hence, with a 5% chance, a given supergame ends after each round. Participants are informed about the continuation probability at the beginning of each supergame. It corresponds to the discount rate of $\delta = 0.95$ that is used for the algorithms in the simulation treatments.[16] To allow for different experimental sessions with the same supergame lengths, the round numbers are pre-drawn with a random number generator.[17] At the end of each round, participants receive information about all prices in the market. Furthermore, they see their own profit in the given round.

[15]Three supergames are usually sufficient to observe learning in similar environments. I provide details on this in Appendix A.1. Furthermore, in Appendix A.5, I report on additional treatments in which I double the number of supergames as a robustness check.
[16]For a risk-neutral player the continuation probability is theoretically equivalent to the discount rate (see for instance Roth and Murnighan, 1978; Dal Bó, 2005).
[17]The exact round numbers are 25 (Supergame 1), 17 (Supergame 2) and 11 (Supergame 3).

Subjects receive the instructions at the start of the session, but they are also available during the experiment at any point. After the participants read the instructions, I ask them a set of control questions.[18] If a participant gives three times a wrong answer, I show an additional explanation for the respective question. One person dropped out of the experiment due to technical problems. I exclude the entire matching group of this subject from the analysis.

Additional details for the experiments with human-algorithm interactions In mixed markets with humans and algorithms, each participant has complete information on which firms use the selected pricing algorithm. While this is arguably not the case in actual markets, Normann and Sternberg (2023) show that participants are insensitive to the knowledge of playing against an algorithm or a human player but that the strategies of the algorithms are driving outcomes.

Following Normann and Sternberg (2023), all profits obtained by a firm that uses a pricing algorithm are given to a passive human player who does not make any active decisions. It rules out any differences in social or distributional preferences that might arise elsewise across treatments.

The framing regarding the algorithmic decisions in the experiment is neutral. Subjects do not know the exact objective function of the algorithm but that it acts in the interest of another participant, which receives the profits that the selected algorithm makes. They do

[18] See Appendix B for the full set of instructions, the control questions, and screenshots of the relevant decision screens.

not know that it is self-learnt or how it learnt its strategy. It mimics the information structure in actual markets where firms are unaware of a competitor's precise pricing algorithm specifications, yet they understand that the algorithm operates in the competitor's interest.

Number of independent observations In total, 313 participants were recruited, with between 60 to 64 subjects in each experimental treatment (see Table 2). I distinguish between active participants, who make the pricing decision themselves, and passive participants who are only paid when an algorithm makes a profit to keep social preferences the same across treatments.

The number of independent observations in the experiment depends on how participants are grouped into matching groups. A matching group is a set of active participants who are matched with one another across supergames, which may lead to potential dependencies in learning and behaviour. Each matching group contains three markets. When there are two active participants per market, the group consists of six participants. In treatments with three active participants per market, the group consists of nine. This setup results in 10 independent matching groups for the 2H0A treatment and 7 for 3H0A

In mixed markets, human participants interact with the selected algorithm that no longer learns but follows the strategy it developed during the simulations. Consequently, there are no reputation effects associated with the selected algorithm across supergames, and the number of independent observations in mixed treatments is determined differently compared to human-only treatments.

In 1H1A and 1H2A, each market includes only one human participant who takes active pricing choices, and there are no reputation effects across supergames. Therefore, the number

of independent observations corresponds to the number of markets (32 for 1H1A and 21 for 1H2A).

In 2H1A, two active participants exist in each market, which can give rise to reputation effects across supergames. There are 42 participants who make active decisions. The other 21 participants are passive and only paid based on the algorithm's decisions. Similarly to 1H1A, the passive participants are excluded from the matching scheme because they do not contribute to any reputation effect. Given a matching group size of six, this treatment has seven independent observations.

For treatments without any humans (0H2A and 0H3A), I use 1,000 independent simulation runs with the main parameterisation of the selected algorithms as described in Section 3.5 as the comparison unit.[19]

Table 2: Number of observations by treatment

Treatment	Number of participants	Number of independent observations
3H0A	63	7
2H0A	60	10
1H1A	64	32
1H2A	63	21
2H1A	63	7
0H2A	-	1,000
0H3A	-	1,000

Experimental implementation and payment The experiments were conducted online in May and June 2021. I recruited the participants using ORSEE (Greiner, 2015) from

[19]The large number of simulation runs ensure that standard errors around average prices are close to zero.

the DICE Lab, University of Düsseldorf subject pool. A web-conference call accompanied each session where participants could ask clarifying questions and receive technical assistance if required.[20] Furthermore, in the conference call, I clarified that the instructions are the same for all participants. The experiment was anonymous, and participants could not communicate with each other.

Each session lasted approximately 30 minutes, and subjects earned, on average, 11.3 Euros, including a show-up fee of 4 Euros. A single supergame was selected at random to determine this payoff. The payoff was the sum of profits within this supergame. Within the experiment, I used an experimental currency unit (ECU) where one Euro corresponded to 130 ECU. The experiment employed a between-subject design; thus, each subject participated only in one treatment. Treatments were randomised on the session level. I programmed the experiment in oTree (Chen *et al.*, 2016a).

4.3 Algorithms in markets with human-algorithm interactions

There are several treatments in which humans interact with pricing algorithms in the same environment. In these "mixed" markets, the algorithms learn through self-play before the experiment. Put differently, they develop their strategies in a simulated market environment, where they compete against other algorithms. In the experiment with humans, the selected algorithms do not learn further from new market information but use the strategy they came up with during training. Thus, I do not consider how humans and algorithms may co-learn during real-time interactions.

[20]The procedure is similar to Zhao *et al.* (2020) or Danz *et al.* (2021). Li *et al.* (2021b) find that this procedure offers comparable results to lab experiments in different economic games.

While this approach might appear restrictive, it is arguably realistic. Q-learning is a slow learning algorithm as it updates only one cell of the Q-matrix in each training step. Also, each cell has to be visited by the agent multiple times to obtain an accurate estimate of the value of this state-action combination. As a result, Q-learning algorithms usually learn too slowly to be trained in the actual market environment.

Crucially, training algorithms through self-play in a simulated environment is standard in reinforcement learning and has been proven effective in different applications. For example, reinforcement learning algorithms that outperformed humans in the board game Go, chess, or complex strategic video games were trained using self-play (Silver *et al.*, 2018, 2017; Vinyals *et al.*, 2019). While these zero-sum games differ from pricing environments, they share important similarities, as, for instance, both involve incentives to optimise for long-run outcomes, development of strategic behaviour, and adapting to opponents' actions.

There are clear benefits from this training approach that likely carry over to pricing environments, which makes it a reasonable choice from an industry perspective. Training algorithms in an environment where they compete against constantly adapting opponents may lead to more robust strategies than training the algorithms from a fixed data set. Algorithms must continuously adapt to the evolving strategies of their competitors. This can encourage the development of new strategies that generalise well to various types of competitors, even ones that behave differently from the training partners at convergence.[21] This is particularly relevant in markets where firms may not know whether their competitors

[21] In the context of the board game of Go, Silver *et al.* (2017) show that reinforcement learning algorithms trained solely by self-play develop new strategies that were previously not used by humans. Furthermore, the algorithm beats a variety of other models in the game.

are humans or algorithms, and it is crucial to use a general enough strategy to perform well against a diverse set of opponents.[22]

Furthermore, a firm deploying a pricing algorithm would want to evaluate its performance before entering the market to mitigate the risk of suboptimal pricing strategies. Training algorithms in self-play ensures that the selected algorithm can be tested against different opponents before it is used in the actual market. For further discussion on training reinforcement learning algorithms in artificial environments, also see Calvano *et al.* (2020b).

As described in Section 3.3, the Q-learning algorithms always condition their current pricing decision on a state, which is the set of prices from the previous period. In the first round of each supergame, the selected algorithm has no state to condition upon as the state $\mathbf{s}_{t=0}$ is undefined. To circumvent this initial condition problem, I define $\mathbf{s}_{t=0}$ as the state of convergence from the learning process. Thus, the selected algorithm always begins each supergame with the same action played last in the simulated environment.

5 Hypotheses

From a theoretical perspective, punishment strategies are vital for collusion to be sustainable in the long run (Friedman, 1971; Abreu, 1988). While humans often fail to employ punishment strategies that appear desirable from the theoretical perspective, Calvano *et al.* (2020b) and Klein (2021) find that self-learnt pricing algorithms learn harsh punishment strategies that make collusion incentive-compatible.[23] I expect the Q-learning algorithms

[22]When training reinforcement learning algorithms in a simulated environment for pricing, the market environment has to be known to the developing firm. However, with modern tools in demand estimation and supervised machine learning, it is reasonable to assume that firms can derive this environment and benefit from the advantages of this learning method.

[23]For a discussion of the strategies that humans use in experimental market games see for instance Wright (2013).

in my design to learn comparable punishment strategies, which would theoretically foster collusion compared to lenient strategies often used by humans.

Algorithms may also reduce the strategic uncertainty within the game. In the mixed market treatments, the selected algorithm plays according to their fixed limit strategy against humans. Normann and Sternberg (2023) argue that playing against a deterministic algorithm reduces strategic uncertainty compared to playing against a human, who might change the strategy during the game and are less committed to a particular behaviour. They demonstrate that the postulated reduction in strategic uncertainty fosters collusion. As the selected algorithm in my design is also deterministic after convergence, I expect higher degrees of tacit collusion in mixed markets relative to human markets. Since humans first have to learn about the selected algorithm's strategy, it appears natural that this effect especially materialises in later supergames. Harsher punishment strategies and reduced strategic uncertainty by algorithms should foster collusion. Thus, I hypothesise that market prices increase as more firms use a pricing algorithm for a given market size.

Hypothesis 1. The level of tacit collusion increases in the share of firms using self-learnt pricing algorithms for a given market size.

It is a well-documented finding in the literature on experimental market games that tacit collusion becomes less likely as the number of firms in the market increases (Engel, 2007; Huck *et al.*, 2004b; Harrington *et al.*, 2016). Within my design, a larger market size implies higher deviation profits. That, in turn, increases the incentive to deviate from a collusive price level. Furthermore, the strategic complexity of the game grows as the number of firms increases. With more firms in the market, market participants have to condition

their behaviour on additional factors, such as the previously chosen prices from the extra competitor. This increase in strategic complexity may further hinder collusion.[24]

Similar to the findings in experimental market games, Calvano *et al.* (2020b) and Johnson *et al.* (2023) find decreasing prices in algorithmic markets in their simulations as the number of firms increases. I expect comparable results in my experimental design, which leads to the following hypothesis.

Hypothesis 2. The level of tacit collusion decreases in the number of firms in the market for human and algorithmic markets alike.

Note that it is unclear how those number effects differ between algorithmic and human markets. While the decline in market prices in the previous studies on algorithmic collusion appears smaller than in human markets, the market setup deviates substantially from the environments usually used in experimental market games. Hence, it is an open question whether algorithms are better at colluding as the market size increases. I investigate this question in the following sections.

To measure the degree of tacit collusion and test the hypothesis, I use the market prices. Within the experimental treatments, an independent observation is the average market price for a given matching group. For the simulations, I average the market prices for each independent simulation over 1,000 rounds after convergence. All hypotheses and the corresponding statistical tests have been pre-registered.[25]

[24]For a discussion on the influence of strategic complexity on cooperation see Jones (2014) and Gale and Sabourian (2005).
[25]The pre-registration uses the template from *AsPredicted.org* and can be found here https://osf.io/yd32b and here https://osf.io/uxdcp. Ethical approval was granted by the German Association for Experimental Economic Research e.V. (No. vzRbKXHq).

6 Results

6.1 Algorithmic and human markets

I present the results from the markets with only humans and only algorithms in Figure 1. For algorithms, I focus on the parameterisation of the selected algorithm for the reasons discussed in Section 3.4.[26] I provide simulation results on the whole grid of α and β in Appendix A.2.

Collusion among algorithms Both in two and three-firm markets, algorithms converge to non-competitive price levels.[27] In 0H2A the average market price across 1,000 independent simulation runs is 3.975 and thus, close to the monopoly price of 4. Also, in 0H3A, the average market price of 2.259 is above p^{NE}. However, prices are substantially lower in 0H3A than in 0H2A. Thus, increasing the market size reduces market prices and leads to more competitive markets. Notably, these results are robust to varying the learning rate α and the exploration decay β (see Appendix A.2). These findings are summarised in Result 1. They align with Hypothesis 2 and previous findings by Calvano *et al.* (2020b).

Importantly, these price levels above p^{NE} are not myopic behaviour, but they are actually supported by punishment strategies that the algorithms learn. The algorithms learn to punish deviations from the collusive price level and make collusion, thereby incentive-compatible in the vast majority of simulation runs. Indeed, for 81.4% of all simulation runs in two-firm

[26] For 0H2A these parameters are $\alpha \approx 0.027$ and $\beta \approx 1 \times 10^{-8}$. The values for 0H3A are $\alpha \approx 0.029$ and $\beta \approx 6.16 \times 10^{-7}$.
[27] While convergence is not guaranteed, the algorithms always converge using the convergence criterion defined in Section 3.3.

markets, algorithms learn a limit strategy that makes deviations unprofitable. In three-firm markets, they do so for 62.7% of all simulation runs.[28]

Result 1 (Algorithmic collusion). Q-learning algorithms learn to play prices above the $p = 1$ stage-game Nash equilibrium. These price levels are supported by punishment strategies. It suggests that algorithms learn to collude.

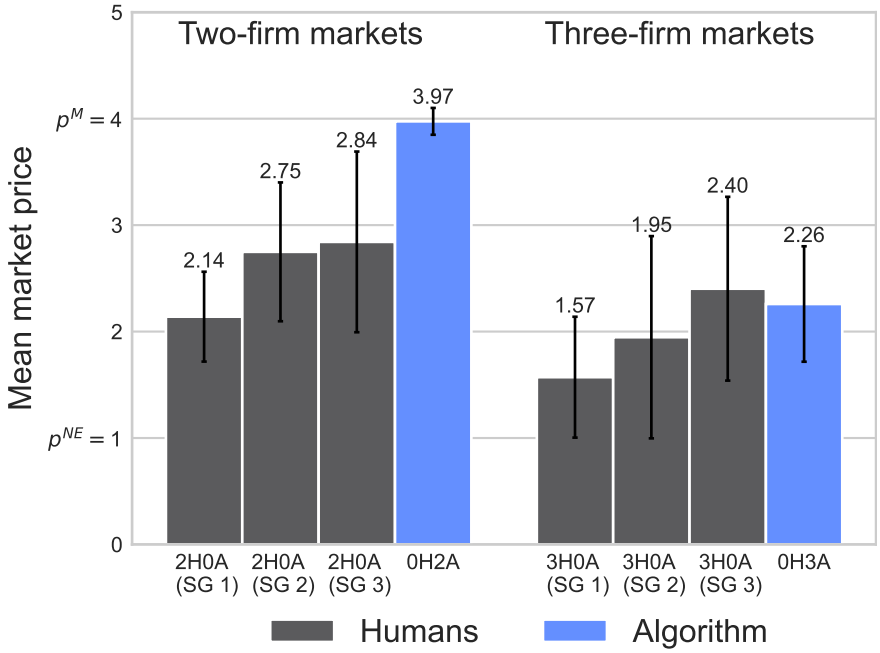

Figure 1: Market prices for the algorithmic and human markets for each supergame (SG). I derive the prices for the algorithmic markets upon convergence as an average over 1,000 subsequent periods for 1,000 independent simulations. The error bars represent the standard deviation.

Collusion among humans Similar to previous findings in the literature (e.g., Huck *et al.*, 2004b), collusion becomes more difficult for humans as the market size increases. Average market prices are higher for each supergame in 2H0A compared to 3H0A. In 2H0A, they

[28] For further details on how I derive these numbers, see Appendix A.4.

increase from 2.140 in the first supergame to 2.749 in the second, and reaching 2.842 in the last supergame. In 3H0A, they move from 1.571 to 1.947 and finally to 2.403. These differences are (weakly) statistically significant for the first and second supergame but insignificant for the last supergame (SG1 $p = 0.045$; SG2 $p = 0.055$, SG3 $p = 0.283$; two-sided Mann–Whitney U tests). Thus, while the difference in prices becomes smaller after learning, prices are always higher in two-firm markets compared to three-firm markets with humans. This finding is in line with Hypothesis 2.

While both algorithms and humans see a drop in price due to the increase in market size, the decline is more substantial for algorithmic markets. While the difference between two and three-firm markets is around 0.439 in the last supergame for humans, it is 1.716 for algorithms. It suggests that increasing the market size hampers algorithmic collusion more than human collusion. As such, fostering larger market sizes can be an even more effective tool for promoting competition in algorithmic markets.

Comparing algorithmic and human collusion In two-firm markets, algorithms outperform humans at colluding. Average market prices in 0H2A are statistically significantly higher than in 2H0A for each supergame ($p < 0.01$ for all supergames; two-sided Mann–Whitney U tests).

In three-firm markets, algorithms are more collusive than humans in the initial supergames. However, this advantage entirely fades after the first two supergames as there are no differences between algorithms and humans in the third and last supergame (SG1 $p < 0.01$; SG2 $p < 0.01$, SG3 $p = 0.980$; two-sided Mann–Whitney U tests). Hence, after humans had the chance to learn about the game, they are as good as self-learnt algorithms

at colluding in three-firm markets. In other words, trained algorithms can outperform inexperienced humans at colluding in markets with three firms. Yet, humans are as good as algorithms at colluding after they gain experience.

Notably, also when comparing the average market prices of humans to other parameterisations of the algorithm, the results are similar (see Appendix A.2 for details). Moreover, in Appendix A.5, I report additional treatment in which I increase the number of supergames for humans from three to six. Likewise, algorithms outperform humans in two-firm markets in these treatments. Also, in three-firm markets, the results remain the same in the sense that there is no statistically significant difference in market prices for any of the supergames after the second one.

Result 2 (Comparing algorithmic and human collusion).

- Market prices are decreasing in the number of firms for humans and algorithmic markets alike.

- Q-learning algorithms are more collusive than humans in two-firm markets.

- In three-firm markets, Q-learning algorithms are more collusive than humans in the first two supergames. Market prices are similar after humans learnt to play the game.

Result 2 summarises the first main findings of this paper. Within a duopoly, algorithmic markets are more collusive than human markets. Hence, algorithmic collusion can increase market prices and hurt competition compared to human pricing. Also, in three-firm markets, algorithms outperform inexperienced humans at colluding. However, in the last supergame,

the average market price in human markets is similar to algorithmic markets. Thus, algorithmic collusion appears less alarming as the market size increases and humans gain experience.

6.2 Algorithmic strategies

Strategy of the selected algorithm in the experiment with humans In the experiment where humans compete against algorithms, I must decide on a specific realisation. Following the discussion in Section 3.4, I use the limit strategy of the algorithm, that maximises the selection criterion Ψ. Hence, it is the best-performing realisation of the algorithm with the same learning rate α and exploration decay β as in Section 6.1.

Equation 6 describes the core idea of the selected algorithm's limit strategy. It is nearly identical for two and three-firm markets.[29]

$$p_i^t(\mathbf{s}^t) = \begin{cases} p^M & \text{if } \mathbf{s}^t = \{p_j^{t-1} = p^M | \forall j\} \\ p^M & \text{if } \mathbf{s}^t = \{p_j^{t-1} = p^{NE} | \forall j\} \\ p^{NE} & \text{otherwise} \end{cases} \tag{6}$$

Upon cooperation at the monopoly price p^M in the previous period, the selected algorithm always chooses the monopoly price again. Any deviation from the cooperative outcome is punished by playing p^{NE}. If and only if all firms played p^{NE} in the previous period, the

[29] There are minor differences between the strategies in 0H2A and 0H3A. Namely, in 0H3A, a small number of states trigger a different response by the selected algorithm after deviations from the monopoly price. For instance the state $\mathbf{s}_t = (4, 3, 0)$ leads to $a_t = 4$ or $\mathbf{s}_t = (4, 4, 2)$ yields $a_t = 3$. However, these states are never reached after the selected algorithms converge. Furthermore, in mixed market experiments, these states only account for approximately 1% of all rounds. Additional details are provided in Appendix A.9.

selected algorithm reverts to playing p^M. The selected algorithm plays p^{NE} in every other relevant state.[30]

Interestingly, this strategy is similar to the win-stay lose-shift strategy (WSLS) discussed by Nowak and Sigmund (1993) in the context of the iterated prisoner's dilemma. Whenever an agent uses WSLS in the iterated prisoner's dilemma, she conditionally cooperates. Upon any deviation, the agent defects and reverts back to cooperation if and only if both players defected in the previous period. WSLS has several desirable properties from an (evolutionary) game-theoretical perspective. If actions are noisy, WSLS can correct for unintended deviations when playing with another agent that uses WSLS. That is not the case for other popular strategies like tit-for-tat. Furthermore, WSLS can detect and exploit unconditional cooperators after unintended deviations, which may arise if the action implementation is noisy. However, agents that always defect can exploit WSLS depending on the exact payoff structure. Nowak and Sigmund (1993) show that WSLS arises naturally as the most widespread strategy in an evolutionary simulation in a noisy iterated prisoner's dilemma and outperforms other strategies.

Notably, the WSLS strategy does not arise by construction but as a result of the algorithm's learning procedure. It demonstrates that training Q-learning algorithms in a simulated environment can lead to strategies like WSLS, which are not only effective in this setting but have been shown to outperform other strategies in strategically similar environments. It highlights that this learning approach encourages the development of strategies known to be among the most successful ones in other settings.

[30]This refers to all possible states that are reachable given the strategy of the selected algorithm. Thus, I do not consider states requiring it to play prices that it never plays itself when following its strategy.

Other algorithmic strategies WSLS is widespread among algorithms, especially among those close to being the optimal one. As I outline in detail in Appendix A.6, I focus on a fixed set of strategies popular in studying collusion among humans and Q-learning algorithms to show this.

I take into account the following strategies: Besides the selected algorithm's strategy, there can be other WSLS variations with different price levels in the collusion and punishment phase. I consider price "cycle" strategies, "always Nash", "myopic" strategies, "tit-for-tat" (TFT), "grim trigger" strategies (GT), and other one-period punishment strategies that coordinate on asymmetric prices in the punishment phase. Furthermore, even with memory-one algorithms, strategies with extended punishment phases after a deviation are possible. Agents could learn them by *jointly* coordinating on an (asymmetric) price vector in a punishment phase.

I consider these strategies for two and three-firm markets for the entire parameter grid of α and β, for the exact parameters of the selected algorithms for different simulation runs, and for algorithms that are close to optimal. For the latter, I consider the 100 best-performing realisations of the algorithm according to the selection criterion Ψ.[31] The intuition is that these algorithms could be chosen by a firm with a similar likelihood as the selected algorithm.

Table 3 presents the results from the strategy classification. In two-firm markets, WSLS is the most frequent strategy for all parameterisations. Across the entire grid, around 34% of algorithms learn a variation of it. This value increases to 39% for algorithms with the same α and β parameters as the selected algorithm. Note that it is not always the exact strategy as described by Equation 6 but variations of WSLS with different collusion and punishment

[31] Note that these may differ in α and β to the actual best-performing algorithm.

Table 3: Proportion of strategies for two and three-firm markets

Strategy	Two firms			Three firms		
	Entire Grid	Parameter Selected	Close to Optimal	Entire Grid	Parameter Selected	Close to Optimal
WSLS (selected)	0.05	0.19	1.00	0.00	0.01	0.26
WSLS (other)	0.29	0.20	0.00	0.04	0.08	0.11
Myopic	0.09	0.02	0.00	0.03	0.38	0.09
Nash	0.00	0.00	0.00	0.53	0.01	0.00
TFT	0.00	0.00	0.00	0.00	0.00	0.00
GT	0.00	0.00	0.00	0.00	0.00	0.00
Cycle	0.06	0.03	0.00	0.14	0.01	0.01
Other one period	0.19	0.39	0.00	0.02	0.06	0.00
Longer punishment	0.31	0.16	0.00	0.24	0.46	0.53
Punishment length in periods						
Avg. length	1.37	1.15	1.00	1.19	1.41	1.75
Avg. length ($t > 0$)	1.52	1.17	1.00	2.70	2.29	1.92

Notes: This table presents the proportion of strategies employed by algorithms in two and three-firm markets for the entire parameter grid, the parameters of the selected algorithm and algorithms that are close to being optimal. A summary and explanation for all strategies is provided in Table A.1 in Appendix A.6. "Avg. length" indicates the average punishment length across all scenarios, while "Avg. length ($t > 0$)" considers only simulations runs with some punishment, excluding "Myopic" and "Always Nash".

levels. When focusing on algorithms that are close to optimal, all learn the exact same WSLS strategy as the selected algorithm.

In three-firm markets, there exists a considerable variation in strategies. Focusing on the entire parameter grid, algorithms mostly learn to play p^{NE}, which accounts for 53% of strategies, while variations of WSLS account for 4% of the data. The share of WSLS increases to 9% for algorithms with the same α and β as the selected algorithm. Intuitively, this share is not higher for these parameters as the average price is only around 2.26. For algorithms that are close to optimal, WSLS is again one of the most frequent strategies, with

26% learning the same strategy as the selected algorithm and another 11% learning other variations of WSLS.

While algorithms never learn grim trigger, they sometimes learn extended punishment strategies, particularly in three-firm markets (53% of near-optimal algorithms), which results in an average punishment length of 1.75 periods. It is not a distinct, well-defined strategy but a set of strategies where the algorithms punish by coordinating on a particular price cycle after a deviation, often with asymmetric prices across firms in the punishment phase.[32] In fact, among near-optimal algorithms in three-firm markets, 81% of these strategies occur only once and follow a unique price path after one of the agents deviates. In other words, these strategies are highly specific to the opponent that the algorithms learnt against.

As highlighted by Eschenbaum *et al.* (2022), Q-learning algorithms in pricing environments can overfit to the strategy of the rival they encountered in the learning environment. It results in a considerable heterogeneity of strategies and limits the ability of the algorithms to extrapolate to new markets with different competitors. Importantly, as discussed in Section 3.5, the selection criterion offers a way to restrict the set of algorithms. It focuses on ones that learn strategies which perform well not only against their training opponent but also against other potential competitors. This is achieved by considering profitability and optimality across all possible states rather than focusing solely on the state of convergence. Consequently, for near-optimal algorithms, the share that learns WSLS increases drastically.

In summary, WSLS is the most common strategy among Q-learning algorithms in two and three-firm markets, particularly for near-optimal ones. This aligns with recent findings by Schaefer (2022); Barfuss and Meylahn (2023); Bertrand *et al.* (2025), and Kasberger

[32] I provide examples for these strategies in Figure A.10 in the appendix.

et al. (2023) for the iterated prisoner's dilemma, that show that WSLS is the most common strategy among Q-learning algorithms. Other strategies in my environment are either non-collusive or overly specific to opponents' behaviours, making WSLS the most reasonable focus for human-machine interactions.

6.3 Collusion between humans and algorithm

Collusion in mixed markets In this section, I consider the outcomes for mixed markets in which humans compete against the selected algorithm with the fixed limit strategy described by Equation 6. Figure 2 shows the average market price pooled across all supergames for all treatments.

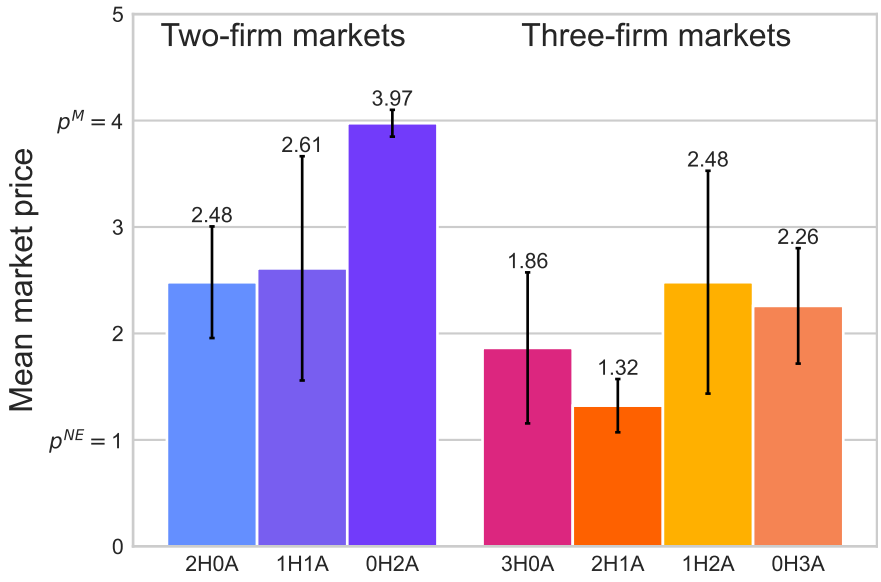

Figure 2: Average market prices for all treatments. For treatment with humans, I pool market prices across all supergames. For algorithmic markets, I use the parameterisation of the selected algorithm as a comparison unit. The error bars represent the standard deviation.

Within two-firm markets, there are no statistically significant differences in market prices between two humans (2H0A) and one human competing with one algorithm (1H1A) ($p = 0.84$, two-sided Mann–Whitney U test). Thus, contrary to Hypothesis 1, the selected algorithm does not foster collusion. Nevertheless, on average, a single selected algorithm is as good at colluding with a human as another human player. Furthermore, prices in 1H1A are significantly lower than in the fully algorithmic market 0H2A ($p < 0.01$, two-sided Mann–Whitney U test). Hence, market prices are weakly increasing in the number of algorithms. While algorithms never foster competition in a duopoly, they make markets more collusive if all firms utilise them.

In three-firm mixed markets, I observe a non-linear relationship between the level of tacit collusion and the number of algorithms in the market. Market prices in 2H1A are *lower* than in 3H0A ($p = 0.07$, two-sided Mann–Whitney U test).[33] Adding another selected algorithm to the market (1H2A) increases prices again compared to 2H1A ($p < 0.01$, two-sided Mann–Whitney U test). There are no statically significant differences between 1H2A and 0H3A ($p = 0.76$, two-sided Mann–Whitney U test). Thus, algorithms only foster collusion if all firms use one, and in particular, only if humans are in the initial two supergames, when they are still inexperienced, as evidenced by a lack of statistically significant market price differences between 3H0A and 0H3A in the last supergame (see Section 6.1).

Heterogeneous strategies in mixed markets In Figure 3, I plot the average market price by round and supergame for each experimental treatment. While 1H2A and 1H1A have a similar trend as 2H0A in the first supergame, 3H0A and 2H1A have noticeably lower

[33]Note that the result differs from Normann and Sternberg (2023) who find that a single tit-for-tat algorithm fosters collusion with three firms in a simpler market environment. The strategies of human sellers drive these results, and I analyse them in the subsequent paragraph.

market prices. In fact, after some initial rounds, average prices in 2H1A are at the $p = 1$ stage game Nash equilibrium. In the later supergames, some interesting patterns emerge after the participants learn about the game. Prices in 2H1A are still close to p^{NE}. While average market prices in 1H1A and 1H2A are similar to 2H0A, there are sharp spikes in every other round.

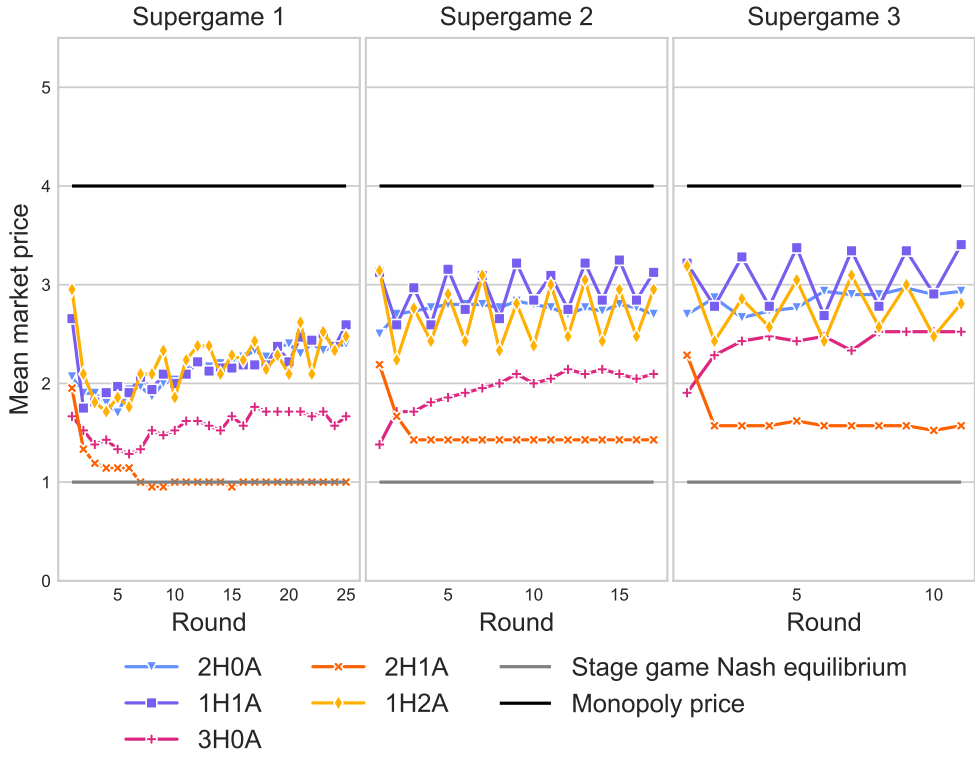

Figure 3: Average market prices by supergame and round for all experimental treatments.

To understand these price patterns, it is essential to remember that participants during the experiment play against the limit strategy of the selected algorithm. In other words, participants play against a variation of a win-stay lose-shift strategy. Normann and Sternberg (2023) demonstrate that the algorithm's strategy is a significant determinant of outcomes in

human-machine interactions. The expectations that participants have about the algorithm's behaviour are mostly irrelevant. Hence, it is essential to understand how participants respond to the selected algorithm's strategy in the presented setup.

While participants do not know the this strategy initially, they can learn about it during the first supergame. Once a participant understands how the selected algorithm works, there are different ways to adapt her strategies, as a response. First, she can ALWAYS COOPERATE with it at the monopoly price. Second, she can ALWAYS DEFECT at the $p = 1$ stage game Nash equilibrium. Moreover, while the limit strategy of the selected algorithm can correct unintended deviations and punish intended deviations by other firms, it is also possible to construct strategies that try to exploit the selected algorithm. To see this, consider a three-firm market where two firms use the strategy described by Equation 6 and firm k uses the following strategy

$$p_k^t(\mathbf{s}^t) = \begin{cases} p^D = 3 & \text{if } \mathbf{s}^t = \{p_j^{t-1} = p^{NE} | \forall j\} \\ p^{NE} & \text{otherwise} \end{cases} \tag{7}$$

Given that the selected algorithm always plays p^M after all firms played p^{NE} in the previous round, the strategy by firm k triggers the selected algorithms to cooperate every other round only to exploit their cooperative phase by choosing the most profitable deviation p^D.

It is straightforward to show that in an infinitely repeated game with $\delta = 0.95$ this EXPLOIT strategy of firm k strictly dominates cooperation at the monopoly price in three-firm markets. However, this strategy is dominated by ALWAYS COOPERATE for two-firm markets (see Appendix A.8 for the details).

Furthermore, they can play an imperfect exploitation strategy by playing a price of $p = 2$ in the cooperative phase of the selected algorithm and $p = 1$ otherwise. I denote this strategy by EXPLOIT2. The strategies ALWAYS DEFECT and EXPLOIT2 are dominated by ALWAYS COOPERATE and EXPLOIT.

To investigate which strategies the participants use in 1H1A and 1H2A against the algorithm, I estimate a mixture model using the Strategy Frequency Estimation Method (SFEM) proposed by Dal Bó and Fréchette (2011). The method is highly influential for estimating strategy choices in infinitely repeated games, especially the prisoner's dilemma (e.g., Fudenberg *et al.*, 2012; Romero and Rosokha, 2018; Dal Bó and Fréchette, 2019). Starting from a predefined set of strategies, SFEM assumes that subject i, chooses strategy $strat^k$ with probability ϕ^k, and follows this strategy for all rounds of the game. In each period, participant i selects her price according to strategy $strat^k$ with probability $\sigma \in (1/2, 1)$ but makes an error with probability $1 - \sigma$. The individual likelihood that participant i plays according to strategy k is given by $P_i(strat^k) = \prod_t \sigma^{I_{t,i}}(1-\sigma)^{1-I_{t,i}}$. The identifier variable $I_{t,i}$ equals 1 if the price of participant i in period t corresponds to the price she would have played if she followed strategy $strat^k$. Otherwise, $I_{t,i}$ equals zero. The log-likelihood function is given by $\mathscr{L} = \sum_i ln(\sum_k \phi^k P_i(strat^k))$. The estimate of ϕ^k represents the share of participants in the population that uses strategy k. The value of σ can be interpreted as a goodness of fit parameter. The model is noisy if σ is close to its lower bound of 0.5. The model describes

the data well for values of σ that are close to 1. For the estimation procedure, I focus on the strategies that are reasonable when competing against the algorithm (ALWAYS COOPERATE, ALWAYS DEFECT, EXPLOIT, and EXPLOIT2). Moreover, I restrict the analysis to the last supergame. Table 4 shows the results of the estimation procedure.

Table 4: Estimated proportion for each strategy

Strategy	Treatment 1H1A	Treatment 1H2A
ALWAYS COOPERATE	0.61 *(0.09)*	0.48 *(0.11)*
ALWAYS DEFECT	0.10 *(0.05)*	0.22 *(0.11)*
EXPLOIT	0.29 *(0.08)*	0.29 *(0.10)*
EXPLOIT2	0.00 *(0.00)*	0.02 *(0.05)*
σ	0.92	0.84

Notes: This table shows the proportions of strategies humans play against the selected algorithm. I estimate them using the strategy frequency estimation method. The mixture model is estimated by maximum likelihood estimation. I restrict the data to the last supergame. The bootstrapped standard errors are in parentheses.

Participants' most frequent strategy against the selected algorithm is ALWAYS COOPERATE in both treatments. The estimated proportion is, however, smaller in 1H2A compared to 1H1A. Also, EXPLOIT is prevalent in the population, but the estimates do not differ between 1H1A and 1H2A. Notably, the share of ALWAYS DEFECT is higher in 1H2A compared to 1H1A. The imperfect exploitative strategy EXPLOIT2 is never played in 1H1A, and it only accounts for a share of 0.02 of the data in 1H2A.

In line with the shift in incentives when increasing the market size, fewer participants play a cooperative strategy against the selelcted algorithm in 1H2A. Yet, participants often

fail to learn the best response as EXPLOIT and ALWAYS COOPERATE dominate ALWAYS DEFECT in 1H2A. A possible reason is that learning about the environment is more difficult in 1H2A due to higher strategic complexity. While both algorithms in 1H2A use a WSLS strategy, participants still have to consider additional information compared to 1H1A. That can impede learning for a subset of participants. Individual prices reveal that some participants circle between prices of 1, 2, and 3 without a clear pattern. It appears that these participants did not learn to follow a fixed strategy (see Appendix A.9 for the price patterns on an individual level). This argument is also supported by the smaller value of σ and higher standard errors in 1H2A, as it indicates a more noisy behaviour of the participants. While average market prices in 1H1A (1H2A) and 2H0A (3H0A) are similar, it is usually not the case for individual markets. Depending on the particular strategies that humans learn, mixed markets can be more or less collusive than their entirely human counterparts.[34]

In 2H1A, it is also crucial to consider the possible strategies humans can use against the algorithm. While ALWAYS COOPERATE and EXPLOIT are both still viable options to play against the selected algorithm, they now require joint coordination by two humans. Indeed, low prices in 2H1A can be explained by a frequent failure to coordinate simultaneously against the algorithm. While some markets manage to collude at the monopoly price, most participants fail to coordinate on any other strategy than ALWAYS DEFECT against the selected algorithm.[35]

Discussion of humans interacting with other limiting strategies A natural question is how humans would compete against other strategies that are not WSLS but which other

[34]Also Hanspach *et al.* (2024) find heterogeneous market outcomes depending on the exact number of algorithms in the market using data from the Dutch online retailer *bol.com*.
[35]Figure A.14 in Appendix A.9 highlights these price patterns.

algorithms occasionally learn. This is particularly relevant for algorithms that are close to optimal according to the selection criterion, as they have a similar likelihood of being chosen by a firm when deciding to deploy one.

There is no variation in strategies among close-to-optimal algorithms in two-firm markets, as discussed in Section 6.2. As such, focusing on the WSLS strategy is largely without any loss of generality. I expect the results from the 1H1A treatment to generalise to other realisations of algorithms.

Similarly, WSLS is a common strategy in three-firm markets. It repeatedly occurs among close-to-optimal algorithms, but algorithms also learn other strategies. In particular, they sometimes adopt specific punishment strategies that can last for more than one period.

In 2H1A, results would likely be similar if firms used algorithms with a longer punishment phase. Participants fail to coordinate on a joint strategy to play against the selected algorithm. This pattern would be similar for different algorithms, especially since coordination on a joint strategy is even more challenging given the overly specific strategies developed in the learning environment against the competitor.

Also, in 1H2A, other close-to-optimal algorithms are unlikely to foster collusion compared to the experiments with the selected algorithm, as the limit strategies are specific to the behaviour of its competitors. At the same, the algorithms tend to punish longer and may not be exploitable. This, in turn, can foster collusion compared to the experiments with the selected algorithm for this market composition.

I conduct additional robustness experiments to investigate the potential pro-collusive effect of other close-to-optimal algorithms in 1H2A. I report them in Appendix A.7. I focus on the algorithms that maximise the selection criterion Ψ, but did not learn WSLS as their limit

strategy. The strategies are more complex and use a longer punishment mechanism. Furthermore, they are not exploitable, and colluding with them at the monopoly price is the best response. Market prices in these additional treatments are smaller than in 1H2A, with WSLS in all supergames. It suggests that the complexity of these non-WSLS strategies can hinder collusion when interacting with humans, even though they are not exploitable. The simpler strategy of WSLS appears beneficial when trying to facilitate collusion in mixed markets with three firms, even when more complex strategies might seem theoretically superior.

Summary and implications Within my framework, firms have a clear incentive to use Q-learning-based pricing algorithms in a duopoly. If only a single firm adopts it, prices do not change. Yet, if both firms outsource their pricing decisions to an algorithm, markets become more collusive, which in turn increases firms' profits.[36] This effect resembles recent findings by Assad *et al.* (2024) on the German gasoline market, showing that prices only increase if both firms in a duopoly adopt a price algorithm.

For triopolies, outcomes depend on the exact market composition. Algorithms only hurt competition if most firms decide to use pricing algorithms and humans lack experience. Furthermore, adoption incentives are less obvious compared to a duopoly, as firms' profits can decrease if only a single firm utilises them. It suggests that algorithmic collusion may, in particular, be a problem in smaller markets where coordination is easier to achieve.

Result 3 summarises the results on all market compositions and the interaction of humans and algorithms.

[36]Köbis *et al.* (2021) argue that the decision to delegate the pricing to an algorithm can be particularly relevant as it allows the firms' manager to morally distance herself from the unethical behaviour of collusion. In my experiment, firms cannot decide whether they want to adopt a pricing algorithm as it is determined exogenously. For experimental evidence on the endogenous decision to delegate to pricing algorithms in oligopoly markets, see Normann *et al.* (2025).

Result 3 (Collusion between humans and algorithms).

- In duopolies, Q-learning algorithms can foster tacit collusion, and market prices are (weakly) increasing in the number of algorithms in the market.
- In triopolies, market outcomes depend on the exact number of Q-learning algorithms in the market. Only if most firms use pricing algorithms and humans lack experience, markets become less competitive.
- Q-learning algorithms learn win-stay lose-shift strategies, and the results in mixed markets differ sharply depending on the strategy that humans use when playing against the selected algorithm.

7 Concluding remarks

This paper investigates the competitive implications of self-learning pricing algorithms. While Calvano *et al.* (2020b) and Klein (2021) document that Q-learning, a type of reinforcement learning algorithm, can learn to be collusive, this study addresses two primary research questions: Firstly, to what extent are self-learning pricing algorithms inherently more collusive than human decision-makers? Secondly, how do these algorithms alter market outcomes when they compete within the same market as humans? To answer these questions, I employ simulations with reinforcement learning algorithms and laboratory experiments with humans. Across different treatments, I vary the market size and the number of firms that use a self-learnt pricing algorithm. This approach allows me to provide a counterfactual for algorithmic collusion for a wide range of possible market compositions. It offers insights into how Q-learning algorithms might foster collusion compared to human markets.

In duopolies, market prices are weakly increasing in the number of firms that use a self-learnt pricing algorithm. Markets with one human and one Q-learning algorithm have similar average market prices compared to entirely human markets. If both firms use a Q-learning pricing algorithm, market prices are close to the monopoly level and significantly higher than in markets with humans. In three-firm markets, market prices decrease if a single firm uses a pricing algorithm. It is driven by the specific strategy the Q-learning algorithms learn and the failure of humans to coordinate with the algorithm. As more firms utilise pricing algorithms, prices increase again in three-firm markets. If all firms use a Q-learning algorithm, market prices can be higher than in human markets. However, the effect fades after humans have the chance to learn about the market environment. The Q-learning algorithms frequently learn a strategy that resembles a win-stay lose-shift strategy. It can make collusion incentive compatible. The outcomes in markets with humans and algorithms depend on the heterogeneous strategies that humans learn to play against the algorithm.

The results highlight the potential anti-competitive effects of self-learning algorithms and their limits. Within the presented framework, the concerns from competition authorities that algorithms can harm the competitive landscape are justified. While the results differ based on the specific market structure, Q-learning algorithms can be more collusive than humans, reducing competition when they dominate the market. This effect is particularly pronounced in duopolies and when all firms employ pricing algorithms. These results align with recent empirical studies (e.g., Assad *et al.*, 2024; Hanspach *et al.*, 2024) while providing a more controlled environment. The considered Q-learning algorithms and the experimental

environment are both simple. Nevertheless, it is plausible that more sophisticated algorithms could achieve similar results and scale to more complex real-world markets.[37]

There are several paths for future research to understand further how algorithms can foster collusion compared to human collusion. First, following the approach from simulation-based (Calvano *et al.*, 2020b) and empirical papers (Assad *et al.*, 2024), this paper focuses on two and three-firm markets as it is the most tractable setup. Yet, my results suggest that algorithms' advantage over humans at colluding reduces as the market size increases. At the same time, collusion between humans becomes increasingly challenging for markets with four or more firms (Huck *et al.*, 2004b). It would be interesting to consider the comparison of human and algorithmic collusion in even larger markets and investigate whether the threat from algorithmic collusion is further reduced as the market size expands.

Following prior research on algorithmic collusion, I focus on situations where firms cannot communicate with each other. It describes the scenario in which algorithms or humans coordinate their prices solely by observing the actions of the other market participants. Experimental evidence suggests that communication drastically fosters collusion among humans as it enhances the coordination possibilities (see, for instance, Fonseca and Normann, 2012). Recent papers Crandall *et al.* (2018) and FAIR *et al.* (2022) suggest that communication can also help algorithms coordinate in strategic situations. Using today's communication tools, such as application programming interfaces, which allow different software applications to interact with each other or shared servers, even algorithms can communicate with each other in market-related situations. It is a compelling path for future research to compare explicit collusion between humans and algorithms.

[37]Hettich (2021) shows that deep reinforcement learning algorithms can be collusive in a more complex market environment.

Another critical concern is how to address algorithmic collusion in terms of its policy implication. My research suggests that, especially in smaller markets, competition authorities would have to act to prevent harm to competition. Current research in computer science focuses on explainable artificial intelligence (see Arrieta *et al.*, 2020). The development objective for these algorithms is that humans can understand their results and the decision process. Also, for pricing algorithms, explainable artificial intelligence is desirable. Understanding why algorithms learn to be collusive and how they must be designed to prevent collusive market outcomes is critical. It would enable companies that seek to steer clear of algorithmic collusion to appropriately design their pricing tools, ensuring compliance with legal and regulatory frameworks. Furthermore, as Calvano *et al.* (2020a) suggest, advancements in explainable artificial intelligence can also provide competition authorities with better tools to possibly audit pricing algorithms, enabling them to assess their potential for collusive behaviour.

Acknowledgments

I thank Emilio Calvano, Joe Harrington, Adrian Hillenbrand, Matthias Hunold, Timo Klein, Nils Köbis, Ulrich Laitenberger, Leonard Treuren, Alexander MacKay, Jeanine Miklós-Thal, Hans-Theo Normann, Andrew Rhodes, Yaroslav Rosokha, Catherine Roux, Christopher Snyder, Yossi Spiegel, and Vasilisa Werner for helpful comments and suggestions. Additionally, I thank Robin Bitter, Leon Heidelbach, and Marlene Merker for excellent research assistance. The pre-registration can be found here https://osf.io/yd32b and here https://osf.io/uxdcp. Ethical approval was granted by the German Association for Experimental Economic Research e. V. (No. vzRbKXHq).

Affiliations

[1]University of Southampton, Southampton, UK

References

Abada, I. and Lambin, X. (2023). 'Artificial intelligence: Can seemingly collusive outcomes be avoided?', *Management Science*, vol. 69(9), pp. 5042–5065.

Abreu, D. (1988). 'On the theory of infinitely repeated games with discounting', *Econometrica*, vol. 56(2), pp. 383–396.

Agarwal, R., Schuurmans, D. and Norouzi, M. (2020). 'An optimistic perspective on offline reinforcement learning', *Proceedings of the International Conference on Machine Learning*, pp. 104–114.

Agrawal, A., Gans, J. and Goldfarb, A. (2019). *The Economics of Artificial Intelligence: An Agenda*, University of Chicago Press.

Arrieta, B.A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R. and Herrera, F. (2020). 'Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI', *Information Fusion*, vol. 58(October 2019), pp. 82–115.

Arulkumaran, K., Deisenroth, M.P., Brundage, M. and Bharath, A.A. (2017). 'Deep reinforcement learning: A brief survey', *IEEE Signal Processing Magazine*, vol. 34(6), pp. 26–38.

Asker, J., Fershtman, C. and Pakes, A. (2024). 'The impact of artificial intelligence design on pricing', *Journal of Economics & Management Strategy*, vol. 33(2), pp. 276–304.

Assad, S., Calvano, E., Calzolari, G., Clark, R., Denicolò, V., Ershov, D., Johnson, J., Pastorello, S., Rhodes, A., Xu, L. and Wildenbeest, M. (2021). 'Autonomous algorithmic

collusion: Economic research and policy implications', *Oxford Review of Economic Policy*, vol. 37(3), p. 459–478.

Assad, S., Clark, R., Ershov, D. and Xu, L. (2024). 'Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market', *Journal of Political Economy*, vol. 132(3), pp. 723–771.

Barfuss, W. and Meylahn, J.M. (2023). 'Intrinsic fluctuations of reinforcement learning promote cooperation', *Scientific reports*, vol. 13(1309).

Bertrand, Q., Duque, J., Calvano, E. and Gidel, G. (2025). 'Q-learners can provably collude in the iterated prisoner's dilemma', *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*.

Borenstein, S. and Shepard, A. (1996). 'Dynamic pricing in retail gasoline markets', *The RAND Journal of Economics*, vol. 27(3), pp. 429–451.

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

Bundeskartellamt and Autorité de la concurrence (2019). 'Algorithms and competition', Published online at `https://www.autoritedelaconcurrence.fr/sites/default/files/algorithms-and-competition.pdf`.

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

Byrne, D.P. and De Roos, N. (2019). 'Learning to coordinate: A study in retail gasoline', *American Economic Review*, vol. 109(2), pp. 591–619.

Calvano, E., Calzolari, G., Denicolò, V., Harrington, J.E. and Pastorello, S. (2020a). 'Protecting consumers from collusive prices due to AI', *Nature*, vol. 370(6520), pp. 1040–1042.

Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2019). 'Algorithmic pricing: What implications for competition policy?', *Review of Industrial Organization*, vol. 55(1), pp. 155–171.

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

Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2021). 'Algorithmic collusion with imperfect monitoring', *International journal of industrial organization*, vol. 79, p. 102712.

Calzolari, G. and Hanspach, P. (2024). 'Pricing algorithms out of the box: A study of the repricing industry', *Available at SSRN 4871394*.

Caro, F. and Gallien, J. (2012). 'Clearance pricing optimization for a fast-fashion retailer', *Operations research*, vol. 60(6), pp. 1404–1422.

Chen, D.L., Schonger, M. and Wickens, C. (2016a). 'oTree-An open-source platform for laboratory, online, and field experiments', *Journal of Behavioral and Experimental Finance*, vol. 9, pp. 88–97, ISSN 22146369, doi:10.1016/j.jbef.2015.12.001.

Chen, J., Xu, Y., Yu, P. and Zhang, J. (2023). 'A reinforcement learning approach for hotel revenue management with evidence from field experiments', *Journal of Operations Management*, vol. 69(7), pp. 1176–1201.

Chen, L., Mislove, A. and Wilson, C. (2016b). 'An empirical analysis of algorithmic pricing on amazon marketplace', *Proceedings of the 25th international conference on World Wide Web*, pp. 1339–1349.

Competition & Markets Authority (2021). 'Algorithms: How they can reduce competition and harm consumers', Published online at `https://www.gov.uk/government/publications/algorithms-how-they-can-reduce-competition-and-harm-consumers`.

Cooprider, J. and Nassiri, S. (2023). 'Science of price experimentation at amazon', *Business Economics*, vol. 58(1), pp. 34–41.

Crandall, J.W., Oudah, M., Tennom, Ishowo-Oloko, F., Abdallah, S., Bonnefon, J.F., Cebrian, M., Shariff, A., Goodrich, M.A. and Rahwan, I. (2018). 'Cooperating with machines', *Nature Communications*, vol. 9(1), pp. 1–12.

Dal Bó, P. (2005). 'Cooperation under the shadow of the future : Experimental evidence from infinitely repeated games', *American Economic Review*, vol. 95(5), pp. 1591–1604.

Dal Bó, P. and Fréchette, G.R. (2011). 'The Evolution of Cooperation in Infinitely Repeated Games: Experimental Evidence', *The American Economic Review*, vol. 101(1), pp. 411–429.

Dal Bó, P. and Fréchette, G.R. (2018). 'On the determinants of cooperation in infinitely repeated games: A survey', *Journal of Economic Literature*, vol. 56(1), pp. 60–114.

Dal Bó, P. and Fréchette, G.R. (2019). 'Strategy choice in the infinitely repeated prisoner's dilemma', *American Economic Review*, vol. 109(11), pp. 3929–3952, ISSN 19447981, doi: 10.1257/aer.20181480.

Danz, D., Gupta, N., Lepper, M., Vesterlund, L. and Winichakul, K.P. (2021). 'Going virtual: A step-by-step guide to taking the in-person experimental lab online', *Available at SSRN 3931028*.

Davies, S., Olczak, M. and Coles, H. (2011). 'Tacit collusion, firm asymmetries and numbers: Evidence from EC merger cases', *International Journal of Industrial Organization*, vol. 29(2), pp. 221–231.

den Boer, A.V., Meylahn, J.M. and Schinkel, M.P. (2024). 'Artificial collusion: Examining supracompetitive pricing by q-learning algorithms', *Amsterdam Law School Research Paper*, (2022-25), pp. 2022–06.

Derakhshan, A., Hammer, F. and Demazeau, Y. (2016). 'Pricecast fuel: Agent based fuel pricing', *Proceedings of the 14th International Conference on Practical Applications of Scalable Multi-Agent Systems (PAAMS 2016)*, pp. 247–250.

Duersch, P., Kolb, A., Oechssler, J. and Schipper, B.C. (2010). 'Rage against the machines: How subjects play against learning algorithms', *Economic Theory*, vol. 43(3), pp. 407–430.

Duffy, J., Hopkins, E. and Kornienko, T. (2024). 'Facing the grim truth: Repeated prisoner's dilemma against robot opponents', .

Engel, C. (2007). 'How much collusion? A meta-analysis of oligopoly experiments', *Journal of Competition Law and Economics*, vol. 3(4), pp. 491–549.

Engel, C. (2015). 'Tacit collusion: The neglected experimental evidence', *Journal of Empirical Legal Studies*, vol. 12(3), pp. 537–577.

Eschenbaum, N., Mellgren, F. and Zahn, P. (2022). 'Robust algorithmic collusion', *arXiv preprint arXiv:2201.00345*.

European Commission (2017). 'Final report on the e-commerce sector inquiry', Published online at `https://ec.europa.eu/commission/presscorner/detail/en/ip_17_1261`.

Ezrachi, A. and Stucke, M.E. (2016). 'Virtual competition', *Journal of European Competition Law & Practice*, vol. 7(9), pp. 585–586, ISSN 2041-7764.

Ezrachi, A. and Stucke, M.E. (2017). 'Artificial intelligence & collusion: When computers inhibit competition', *University of Illinois Law Review*, vol. 2017(5), pp. 1775–1810, ISSN 02769948.

FAIR, Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H. *et al.* (2022). 'Human-level play in the game of diplomacy by combining language models with strategic reasoning', *Science*, vol. 378(6624), pp. 1067–1074.

Fonseca, M.A. and Normann, H.T. (2012). 'Explicit vs. tacit collusion-The impact of communication in oligopoly experiments', *European Economic Review*, vol. 56(8), pp. 1759–1772, ISSN 00142921.

Friedman, J.W. (1971). 'A non-cooperative equilibrium for supergames', *Review of Economic Studies*, vol. 38(1), pp. 1–12, ISSN 1467937X.

Fudenberg, D., Rand, D.G. and Dreber, A. (2012). 'Slow to anger and fast to forgive: Cooperation in an uncertain world', *American Economic Review*, vol. 102(2), pp. 720–749, ISSN 00028282.

Gale, D. and Sabourian, H. (2005). 'Complexity and competition', *Econometrica*, vol. 73(3), pp. 739–769, ISSN 00129682.

Greiner, B. (2015). 'Subject pool recruitment procedures: organizing experiments with ORSEE', *Journal of the Economic Science Association*, vol. 1(1), pp. 114–125, ISSN 2199-6776, doi:10.1007/s40881-015-0004-4.

Hansen, K., Misra, K. and Pai, M. (2020). 'Algorithmic Collusion: Supra-competitive Prices via Independent Algorithms', *Marketing Science*, vol. 40(1), pp. 1–12.

Hanspach, P., Sapi, G. and Wieting, M. (2024). 'Algorithms in the marketplace: An empirical analysis of automated pricing in e-commerce', *Information Economics and Policy*, vol. 69, p. 101111.

Harrington, J.E. (2018). 'Developing competition law for collusion by autonomous artificial agents', *Journal of Competition Law and Economics*, vol. 14(3), pp. 331–363, ISSN 17446422.

Harrington, J.E. (2022). 'The effect of outsourcing pricing algorithms on market competition', *Management Science*, vol. 68(9), pp. 6889–6906.

Harrington, J.E., Hernan Gonzalez, R. and Kujal, P. (2016). 'The relative efficacy of price announcements and express communication for collusion: Experimental findings', *Journal*

of Economic Behavior and Organization, vol. 128(051), pp. 251–264, ISSN 01672681, doi: 10.1016/j.jebo.2016.05.014.

Hettich, M. (2021). 'Algorithmic collusion: Insights from deep learning', .

Horstmann, N., Krämer, J. and Schnurr, D. (2018). 'Number Effects and Tacit Collusion in Experimental Oligopolies', *Journal of Industrial Economics*, vol. 66(3), pp. 650–700, ISSN 14676451, doi:10.1111/joie.12181.

Huck, S., Normann, H.T. and Oechssler, J. (1999). 'Learning in cournot oligopoly–an experiment', *The Economic Journal*, vol. 109(454), pp. 80–95.

Huck, S., Normann, H.T. and Oechssler, J. (2004a). 'Through trial and error to collusion', *International Economic Review*, vol. 45(1), pp. 205–224.

Huck, S., Normann, H.T. and Oechssler, J. (2004b). 'Two are few and four are many: Number effects in experimental oligopolies', *Journal of Economic Behavior and Organization*, vol. 53(4), pp. 435–446, ISSN 01672681.

Jeschonneck, M. (2021). 'Collusion among autonomous pricing algorithms utilizing function approximation methods', .

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

Jones, M.T. (2014). 'Strategic complexity and cooperation: An experimental study', *Journal of Economic Behavior and Organization*, vol. 106, pp. 352–366, ISSN 01672681.

Kasberger, B., Martin, S., Normann, H.T. and Werner, T. (2023). 'Algorithmic cooperation', *Available at SSRN 4389647*.

Klein, T. (2021). 'Autonomous algorithmic collusion: Q-learning under sequential pricing', *The RAND Journal of Economics*, vol. 52(3), pp. 538–558.

Köbis, N., Bonnefon, J.F. and Rahwan, I. (2021). 'Bad machines corrupt good morals', *Nature Human Behaviour*, vol. 5(6), pp. 679–685, ISSN 23973374.

Kühn, K.U. and Tadelis, S. (2017). 'Algorithmic collusion', *Prepared for CRESSE 2017*.

Kunz, M., Birr, S., Raslan, M., Ma, L. and Januschowski, T. (2023). 'Deep learning based forecasting: A case study from the online fashion industry', *In: Forecasting with Artificial Intelligence: Theory and Applications*, pp. 279–311.

Leisten, M. (2024). 'Algorithmic competition, with humans', Version from May 9, 2024. Available at `https://www.researchgate.net/publication/349681786_Algorithmic_Competition_with_Humans`.

Lerer, A. and Peysakhovich, A. (2017). 'Maintaining cooperation in complex social dilemmas using deep reinforcement learning', *arXiv preprint arXiv:1707.01068*.

Li, H., Simchi-Levi, D., Sun, R., Wu, M.X., Fux, V., Gellert, T., Greiner, T. and Taverna, A. (2021a). 'Large-scale price optimization for an online fashion retailer', *In: Innovative Technology at the Interface of Finance and Operations, Volume II*, pp. 191–224.

Li, J., Leider, S., Beil, D.R. and Duenyas, I. (2021b). 'Running online experiments using web-conferencing software', *Journal of the Economic Science Association*, vol. 7(2), pp. 167–183.

Littman, M.L. (1994). 'Markov games as a framework for multi-agent reinforcement learning', *In: Machine Learning Proceedings 1994*, pp. 157–163.

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

Loh, E., Khandelwal, J., Regan, B. and Little, D.A. (2022). 'Promotheus: An end-to-end machine learning framework for optimizing markdown in online fashion e-commerce', *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3447–3457.

Madeka, D., Torkkola, K., Eisenach, C., Luo, A., Foster, D.P. and Kakade, S.M. (2022). 'Deep inventory management', *arXiv preprint arXiv:2210.03137*.

March, C. (2021). 'Strategic interactions between humans and artificial intelligence: Lessons from experiments with computer players', *Journal of Economic Psychology*, vol. 87.

Mehra, S.K. (2016). 'Antitrust and the robo-seller: Competition in the time of algorithms', *Minnesota Law Review*, vol. 100(4), pp. 1323–1375, ISSN 00265535.

Mehrotra, P., Pang, L., Gopalswamy, K., Thangali, A., Winters, T., Gupte, K., Kulkarni, D., Potnuru, S., Shastry, S. and Vuyyuri, H. (2020). 'Price investment using prescriptive analytics and optimization in retail', *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3136–3144.

Mengel, F. (2018). 'Risk and temptation: A meta-study on prisoner's dilemma games', *The Economic Journal*, vol. 128(616), pp. 3182–3209.

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

Miller, N.H. and Weinberg, M.C. (2017). 'Understanding the Price Effects of the MillerCoors Joint Venture', *Econometrica*, vol. 85(6), pp. 1763–1791.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015). 'Human-level control through deep reinforcement learning', *Nature*, vol. 518(7540), pp. 529–533, ISSN 14764687.

Musolff, L. (2022). 'Algorithmic pricing facilitates tacit collusion: Evidence from e-commerce', .

Normann, H.T., Rulié, N., Stypa, O. and Werner, T. (2025). 'Delegate pricing decisions to an algorithm? experimental evidence', .

Normann, H.T. and Sternberg, M. (2023). 'Human-algorithm interaction: Algorithmic pricing in hybrid laboratory markets', *European Economic Review*, vol. 152, p. 104347.

Nowak, M. and Sigmund, K. (1993). 'A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game', *Nature*, vol. 364(6432), pp. 56–58, ISSN 00280836.

O'Connor, J. and Wilson, N.E. (2021). 'Reduced demand uncertainty and the sustainability of collusion: How AI could affect competition', *Information Economics and Policy*, vol. 54(100882), ISSN 01676245.

Qin, Z.T., Zhu, H. and Ye, J. (2022). 'Reinforcement learning for ridesharing: An extended survey', *Transportation Research Part C: Emerging Technologies*, vol. 144, p. 103852.

Romero, J. and Rosokha, Y. (2018). 'Constructing strategies in the indefinitely repeated prisoner's dilemma game', *European Economic Review*, vol. 104, pp. 185–219, ISSN 00142921.

Roth, A.E. and Murnighan, J.K. (1978). 'Equilibrium behavior and repeated play of the prisoner's dilemma', *Journal of Mathematical Psychology*, vol. 17(2), pp. 189–198, ISSN 10960880.

Schaefer, M. (2022). 'On the emergence of cooperation in the repeated prisoner's dilemma', *arXiv preprint arXiv:2211.15331*.

Schwalbe, U. (2018). 'Algorithms, machine learning, and collusion', *Journal of Competition Law & Economics*, vol. 14(4), pp. 568–607, ISSN 1744-6414.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Driessche, G.V.D., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M. and Kavukcuoglu, K. (2016). 'Mastering the game of Go with deep neural networks and tree search', *Nature*, vol. 529(7585), pp. 484–489, ISSN 0028-0836.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T. *et al.* (2018). 'A general reinforcement learning algorithm that masters chess, shogi, and go through self-play', *Science*, vol. 362(6419), pp. 1140–1144.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T. and Hassabis, D. (2017). 'Mastering the game of Go without human knowledge', *Nature*, vol. 550(7676), pp. 354–359, ISSN 14764687, doi:10.1038/nature24270.

Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P. *et al.* (2019). 'Grandmaster level in starcraft ii using multi-agent reinforcement learning', *Nature*, vol. 575(7782), pp. 350–354.

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

Wang, Q., Huang, Y., Singh, P.V. and Srinivasan, K. (2023). 'Algorithms, artificial intelligence and simple rule based pricing', *Available at SSRN 4144905*.

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

Watkins, C.J.C.H. and Dayan, P. (1992). 'Q-Learning', *Machine Learning*, vol. 8, pp. 279–292.

Wright, J. (2013). 'Punishment strategies in repeated games: Evidence from experimental markets', *Games and Economic Behavior*, vol. 82, pp. 91–102, ISSN 08998256.

Zhao, S., López Vargas, K., Friedman, D. and Gutierrez, M. (2020). 'Ucsc leeps lab protocol for online economics experiments', *Available at SSRN 3594027*.