TI 2022-067/VII
Tinbergen Institute Discussion Paper

Artificial Collusion: Examining Supracompetitive Pricing by Q-learning Algorithms

Arnoud V. den Boer[1]
Janusz M. Meylahn[2]
Maarten Pieter Schinkel[1,3]

1 University of Amsterdam

2 University of Twente

3 Tinbergen Institute

Artificial Collusion: Examining Supracompetitive Pricing by Q-learning Algorithms

Arnoud V. den Boer, Janusz M. Meylahn, Maarten Pieter Schinkel*

December 12, 2022

Abstract

We examine recent claims that a particular Q-learning algorithm used by competitors 'autonomously' and systematically learns to collude, resulting in supracompetitive prices and extra profits for the firms sustained by collusive equilibria. A detailed analysis of the inner workings of this algorithm reveals that there is no immediate reason for alarm. We set out what is needed to demonstrate the existence of a colluding price algorithm that does form a threat to competition.

JEL-codes: C63, L13, L44, K21. Keywords: collusion, Q-learning, algorithm, pricing

1 Introduction

There is widespread concern amongst competition specialists and authorities around the globe that the increasing use by businesses of data-driven algorithms to determine operational decisions such as pricing may present increased risks of collusion amongst competitors. The worry is concretely that these algorithms may learn to collude without needing explicit coordination and communication (Ezrachi and Stucke, 2016; Mehra, 2016; OECD, 2017; Ezrachi and Stucke, 2020; Mehra, 2021). That such a 'meeting of the artificially intelligent minds' would escape the existing cartel prohibition and enforcement has become inspiration for various proposals to change competition laws and antitrust enforcement priorities (Harrington, 2018; Gal, 2019; Beneke and Mackenrodt, 2020; Bernhardt and

*den Boer: University of Amsterdam (Korteweg de Vries Institute and Amsterdam Business School), boer@uva.nl. Meylahn: University of Twente (Department of Applied Mathematics), j.m.meylahn@utwente.nl. Schinkel: University of Amsterdam (Department of Economics) and Tinbergen Institute, m.p.schinkel@uva.nl. We thank Ali Aouad, John Asker, Vincenzo Denicolò, Justin Johnson, Steve Tadelis, and Ulrich Schwalbe for comments that helped us to improve an earlier version of this paper, Ibrahim Abada and Xavier Lambin for providing useful references, and Joe Harrington, Timo Klein, and Rein Wesseling for helpful discussions on the subject.

Dewenter, 2020; Coglianese and Lai, 2022; Gal, 2022; Mazundar, 2022). Algorithmic collusion has become a call to arms for antitrust authorities (Calvano et al., 2020*b*; Assad et al., 2021), that does not go unheeded (CMA, 2021). Cases of raised prices are being reported in online platforms where algorithmic sellers bid against each other, such as Bol.com (Wieting and Sapi, 2021) and Amazon (Brown and MacKay, 2021). The business community is being warned for liability risks of possible rogue collusion by pricing algorithms (Bertini and Koenigsberg, 2021). Others, however, assert that collusion between pricing algorithms would be hard to sustain, even 'science fiction', and the enforcement attention for it is a waste of resources (Schwalbe, 2019; Veljanovski, 2022; Kühn and Tadelis, 2018). Understanding what threat algorithmic collusion may pose to competition clearly is an urgent and important matter.

A rapidly expanding literature considers the challenge of finding a pricing algorithm that learns to collude without coordination. An influential contribution is Calvano et al. (2020*a*), in which a Q-learning algorithm is said to 'autonomously' learn to collude when both firms in a duopoly use this algorithm. Collusive properties of similar pricing algorithms have also been reported in various other settings, including sequential-move games (Klein, 2021), sellers on a platform (Sánchez-Cartas and Katsamakas, 2022), settings with more advanced reinforcement learning methods (Hettich, 2021; Kastius and Schlosser, 2021; Wang, 2022), play against simple pricing rules (Wang, Huang and Singh, 2022), dealer markets with multiple market makers (Han, 2022; Xiong and Cont, 2021), and continuous-time models (Cartea et al., 2022).

In this paper we critically examine autonomous algorithmic collusion on the basis of claims made by Calvano et al. (2020*a*) that their Q-learning algorithm systematically learns to play collusive strategies, resulting in sizable extra-profits and supracompetitive prices sustained by collusive strategies. The algorithm is claimed to truly collude rather than sustain higher prices 'by mistake', because the learned equilibrium strategies are said to involve a reward-punishment scheme that incentivizes firms 'to consistently price above the competitive level' (p. 3269). The authors write that their conclusions are tentative but their findings nevertheless 'do suggest that algorithmic collusion is more than a remote theoretical possibility' (p. 3268) and should 'ring an alarm bell' (p. 3295) with competition authorities.

To examine these concerns, we provide a detailed explanation of the inner workings of the Q-learning algorithm studied by Calvano et al. (2020*a*) in a tractable setting with two feasible prices. This allows us to characterize the existence and nature of collusive equilibria, defined as strategy pairs that can be interpreted as having a reward-punishment scheme. We identify why convergence to these equilibria is exceedingly slow and often unsuccessful. Moreover, we show by numerical simulations that in this setting the firms

do *not* systematically learn to play collusive strategies, that a substantial amount of supracompetitive limit prices is *not* generated by collusive equilibria, and that changing the hyperparameters of the algorithm does not resolve the problem.

We also identify a problem with the argument used to infer that their algorithms converge to collusive equilibria. The authors observe a particular price pattern after a forced price cut, interpret this as a 'reward-punishment scheme', and conclude that supra-competitive prices are sustained by collusive equilibria. We show, however, that the same price pattern can be generated by non-collusive strategies. Examination of the performance measure used by Calvano et al. (2020*a*) to quantify the benefits or magnitude of collusion furthermore reveals how increases in the firms' objective are caused by factors unrelated to collusion. We also argue that the concept of collusive strategy equilibria is not an appropriate equilibrium concept, but that algorithms should rather be evaluated in terms of the firms' actual objective. By numerically comparing the performance of Q-learning against a reasonable alternative, we show that Q-learning may perform very poorly and is far out of equilibrium.

Our examination of the evidence leads us to conclude that this Q-learning algorithm gives no reason for alarm. We do not believe that, on the basis of the presented simulations, it can be concluded that Q-learning autonomously and systematically learns to collude, nor that the algorithm generates supracompetitive prices sustained by collusive equilibria, nor that this generates extra profits for the firms. Our findings also apply to other papers referred to above where they rely on similar Q-learning algorithms. The challenge to identify a pricing algorithm that learns to collude without coordination has not been met by Q-learning. Constructing well-performing pricing algorithms that learn to collude should certainly be possible, given the many algorithmic techniques that in the last decade have been developed in the dynamic-pricing-and-learning literature within Operations Research. It is therefore commendable and realistic that competition authorities devote resources in order to understand potentially colluding pricing algorithms. Yet constructing a well-performing colluding pricing algorithm is a complex matter, and requires a sophisticated algorithm with properties and performance guarantees that a simple Q-learning algorithm does not have. We therefore propose a concrete set of requirements that future claims of algorithmic collusion should satisfy at the minimum.

Other papers also voice criticism about aspects of algorithmic collusion. Kühn and Tadelis (2018) stress the difficulty of achieving price coordination, which they argue algorithms cannot overcome. Schwalbe (2019) criticizes the simplicity of the algorithm as the source of seeming collusion. Abada and Lambin (2020) are critical about the one-period price cuts used by Calvano et al. (2020*a*) to infer reward-and-punishment schemes and suggest insufficient exploration as one of the drivers of what the authors call 'seemingly collusive

outcomes'. Abada, Lambin and Tchakarov (2022) report that algorithmic sophistication induces defection to competitive prices rather than more robust price elevation. Epivent and Lambin (2022) find pricing patterns in similar algorithmic settings that suggest failure to compete rather than collusion. Eschenbaum, Mellgren and Zahn (2022) criticize offline training and find that collusion breaks down when collusive reinforcement learning policies are extrapolated from a training environment to the market. Asker, Fershtman and Pakes (2021, 2022*a*,*b*) highlight how obtaining supracompetitive limit prices strongly depends on the learning protocol of the algorithms. In particular does the faster synchronous learning protocol, in which the algorithms account for counterfactual earnings of alternative actions, lead to competitive instead of supracompetitive prices. These findings show how sensitive supra-competitive outcomes are to just a little more intelligence.

The rest of this paper is organized as follows. In the next section we summarize the model and the main findings in Calvano et al. (2020*a*). In Section 3 we characterize collusive strategy-equilibria in Q-learning with two feasible prices, we explain why convergence to such collusive equilibria is slow and often fails, and we show that a substantial amount of supracompetitive limit prices are not generated by collusive equilibria. We also show that ad-hoc changes to the algorithm do not resolve this problem, and we explain why the argument used to infer convergence to collusive equilibria is not correct. In Section 4 we explain problems with the performance measure used to quantify collusion, and we show that factors unrelated to collusion can increase the firms' objective. In Section 5 we discuss appropriate and inappropriate equilibrium concepts, and we show that Q-learning can be outperformed by a reasonable and simple alternative. Section 6 summarizes our critique and explains what is needed to demonstrate algorithmic collusion that does form a threat in practice.

2 Model and algorithm

Consider a market environment with two competing firms, labelled 1 and -1, selling substitute products, each firm one. Time is discrete and indexed by $t \in \mathbb{N}$. At the beginning of each time period $t \in \mathbb{N}$, each firm $i = \pm 1$ selects an action (its selling price) $p_i(t) \in \mathcal{A}_i$ from a discrete, non-empty, and finite set of feasible prices $\mathcal{A}_i \subset [0, \infty)$. Subsequently, each firm observes their own demand $d_i(p_i(t), p_{-i}(t))$, where $d_i : [0, \infty)^2 \to [0, \infty)$ is a function unknown to the firms, and earns instantaneous reward $\pi_i(p_i(t), p_{-i}(t))$, where $\pi_i(p_i, p_{-i}) := (p_i - c_i) \cdot d_i(p_i, p_{-i})$ and $c_i \geq 0$ denotes marginal costs. There are no demand shocks. The demand functions $d_i(\cdot, \cdot)$ are according to a multinomial logit model. That is, there are parameters $(a_{-1}, a_1, \mu) \in \mathbb{R}^2 \times (0, \infty)$, unknown to the firms,

such that for both $i = \pm 1$,

$$d_i(p_i, p_{-i}) = \frac{\mathrm{e}^{\frac{a_i - p_i}{\mu}}}{1 + \mathrm{e}^{\frac{a_1 - p_1}{\mu}} + \mathrm{e}^{\frac{a_{-1} - p_{-1}}{\mu}}}, \tag{1}$$

for all nonnegative price pairs.[1]

Each firm's objective is to maximize its cumulative expected discounted payoff stream,

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \delta^t \pi_i(p_1(t), p_2(t))\right], \tag{2}$$

where $\delta \in (0, 1)$ is a common discount factor.

2.1 Description of the algorithm

Both firms use a Q-learning algorithm to determine the prices of their products from the action space. Essentially, Q-learning keeps track of a state variable $s_i(t)$ which consists of the prices charged in the previous time period: $s_i(t) = (p_i(t-1), p_{-i}(t-1))$, for all $t \in \mathbb{N}_{\geq 2}$.[2] The corresponding (finite) state space is equal to $\mathcal{S}_i := \mathcal{A}_i \times \mathcal{A}_{-i}$, for $i = \pm 1$. At the end of each time period $t \in \mathbb{N}$ (i.e., after rewards have been observed), the Q-learning algorithm of firm i computes a so-called Q-matrix $Q^{(i)}(t) = \{Q^{(i)}_{s,p}(t)\}_{s \in \mathcal{S}_i, p \in \mathcal{A}_i}$ via the recursion

$$Q^{(i)}_{s,p}(t) = Q^{(i)}_{s,p}(t-1) \text{ for all } (s,p) \in \mathcal{S}_i \times \mathcal{A}_i \text{ with } (s, p_i) \neq (s_i(t), p_i(t)),$$

and

$$\begin{aligned} Q^{(i)}_{s_i(t),p_i(t)}(t) &= (1-\alpha) Q^{(i)}_{s_i(t),p_i(t)}(t-1) \\ &\quad + \alpha \Big[\pi_i(p_i(t), p_{-i}(t)) + \delta \max_{p \in \mathcal{A}_i} \{Q^{(i)}_{s_i(t+1),p}(t-1)\}\Big]. \end{aligned}$$

Here $\alpha \in [0, 1]$ is a parameter called the *learning rate* and

$$Q^{(i)}(0) = \{Q^{(i)}_{s,p}(0)\}_{s \in \mathcal{S}_i, p \in \mathcal{A}_i}$$

is an initial Q-matrix. Both α and $Q^{(i)}(0)$ are input to the algorithm. The numbers $Q^{(i)}_{s,p}(t)$ are called Q-values corresponding to state s and price p. The Q-matrices are used to determine the prices. It is assumed that both firms apply a so-called ϵ_t*-greedy* type

[1] Calvano et al. (2020*a*) replace the 1 in the denominator of (1) by $e^{a_0/\mu}$. But since the demand function does not change if a constant is added to (a_{-1}, a_0, a_1), we can assume without loss of generality that $a_0 = 0$.

[2] Calvano et al. (2020*a*) first formulate a model with states consisting of the k most recent prices, but then focus in the rest of the paper on the case $k = 1$.

of Q-learning algorithm with $\epsilon_t := \exp(-\beta t)$ for all $t \in \mathbb{N}$ and some $\beta \geq 0$. This means that, for each $t \in \mathbb{N}$ and $i = \pm 1$, price $p_i(t)$ is selected uniformly at random from $\mathcal{A}_i$ with probability ϵ_t; with probability $1 - \epsilon_t$, price $p_i(t)$ is selected in order to maximize the Q-value for the current state:

$$p_i(t) \in \arg\max_{p \in \mathcal{A}_i} Q^{(i)}_{s_i(t),p}(t-1),$$

with ties broken uniformly at random. The parameter β is called the *experimentation parameter*.

2.2 Market environment

Numerical experiments are conducted based on a 'baseline parametrization' where marginal costs are $c_i = 1$ for both firms, and where the demand parameters are set to $(a_{-1}, a_1, \mu) = (2, 2, 1/4)$. The common discount factor is $\delta = 0.95$.

The feasible price sets are constructed as follows. First, the (unconstrained) Bertrand–Nash equilibrium prices (p_1^N, p_{-1}^N) of the one-shot pricing game, and the (unconstrained) prices (p_1^M, p_{-1}^M) that maximize joint profit, are computed. These prices are given by

$$(p_1^N, p_{-1}^N) := \left(\frac{\mu}{1 - V(\mathrm{e}^{(a_1 - c_1)/\mu - 1}\tilde{Q}_0)} + c_1, \frac{\mu}{1 - V(\mathrm{e}^{(a_{-1} - c_{-1})/\mu - 1}\tilde{Q}_0)} + c_{-1} \right),$$
$$(p_1^M, p_{-1}^M) := \Big(\mu(1 + W(A_0)) + c_1, \mu(1 + W(A_0)) + c_1 \Big),$$

where: $V(x)$ is the unique solution v in $(0,1)$ of $v \cdot \exp(v/(1-v)) = x$, $\tilde{Q}_0$ is the unique solution Q_0 to $Q_0 + V(Q_0 \mathrm{e}^{(a_1 - c_1)/\mu}) + V(Q_0 \mathrm{e}^{(a_{-1} - c_{-1})/\mu}) = 1$, $W(x)$ is the Lambert W function which solves $W(x) \cdot \mathrm{e}^{W(x)} = x$, and $A_0 := \exp((a_1 - c_1)/\mu - 1) + \exp((a_{-1} - c_{-1})/\mu - 1)$.[3]

The baseline parametrization is fully symmetric: $c_1 = c_{-1}$ and $a_1 = a_{-1}$, resulting in Nash prices $p^N := p_1^N = p_{-1}^N = 1.4729$ with corresponding payoff $\pi^N := \pi_i(p_i^N, p_{-i}^N) = 0.2229$, and monopoly prices $p^M := p_1^M = p_{-1}^M = 1.9250$, with corresponding payoff $\pi^M := \pi_i(p_i^M, p_{-i}^M) = 0.3375$. Hence, in perfect collusion, in which both firms charge the monopoly price, the cartel increases profit with 51 percent. Note that there is positive profit in competition due the assumption of a logit demand function.

For some integer $m \geq 2$ and some $\xi \in [0, \max_{i=\pm 1} \frac{p_i^N}{p_i^M - p_i^N}]$, the feasible price sets are

[3] For a derivation of these expressions we refer to Li and Huh (2011, Theorem 4 and Theorem 2).

defined by

$$\mathcal{A}_i := \Big\{ p_i^N - \xi(p_i^M - p_i^N) + k \cdot (1+2\xi)\frac{(p_i^M - p_i^N)}{m-1} : k = 0, \ldots, m-1 \Big\},$$

for both players $i = \pm 1$. For $m = 2$ and $\xi = 0$ this results, for example, in the feasible price set $\mathcal{A}_i = \{p_i^N, p_i^M\}$. For higher values of ξ, the firms get more pricing options below the competitive and above the monopoly price. Calvano et al. (2020*a*)'s $m = 15$ and $\xi = 0.1$, for both players, result in the feasible price set

$$\mathcal{A}_i := \{1.4277, 1.4277 + 0.0387, 1.4277 + 2 \times 0.0387, \ldots, 1.9702\},$$

so that the lowest is slightly (3 percent) below the Nash price and the highest feasible price is slightly (2 percent) above the monopoly price.

The initial Q-matrix for both firms $i = \pm 1$ is set to

$$Q_{s,p}^{(i)}(0) = \frac{1}{(1-\delta)}\frac{1}{|\mathcal{A}_{-i}|}\sum_{p_{-i} \in \mathcal{A}_{-i}} \pi_i(p_i, p_{-i}).$$

Numerical experiments are conducted for all possible pairs

$$\begin{aligned}(\alpha, \beta) \in \Psi := \{\alpha_{\min} + \frac{i \cdot (\alpha_{\max} - \alpha_{\min})}{99} : i = 0, 1, \ldots, 99\} \\ \times \{\beta_{\min} + \frac{j \cdot (\beta_{\max} - \beta_{\min})}{99} : j = 0, 1, \ldots, 99\},\end{aligned}$$

with $\alpha_{\min} = 0.025$, $\alpha_{\max} = 0.25$, $\beta_{\min} = 0$, and $\beta_{\max} = 2 \times 10^{-5}$.

2.3 Performance metric

For each $(\alpha, \beta) \in \Psi$, 1000 simulations are conducted in which the two Q-learning algorithms play against each other. In Calvano et al. (2020*a*), a simulation is said to converge if there is a $t_0 \in \mathbb{N}$ with $t_0 \leq 10^9 - 10^5$ and actions $\{a_i(s) \in \mathcal{A}_i : i = \pm 1, s \in \mathcal{S}_i\}$ such that $a_i(s) = \arg\max_{p \in \mathcal{A}_i} Q_{s,p}^{(i)}(t)$ for all $t = t_0 + 1, \ldots, t_0 + 10^5$ and all $i = \pm 1$, $s \in \mathcal{S}_i$. That is, there is 'convergence' if the optimal actions corresponding to all firms and states are unique and remain constant during 100,000 consecutive time periods, within the first billion time periods.

If a simulation converges, Calvano et al. (2020*a*) measure the 'extra-profit compared to the static Nash equilibrium', abbreviated as 'average profit gain' (p. 3277), corresponding to the simulation by

$$\Delta := \frac{\bar{\pi} - \pi^N}{\pi^M - \pi^N},$$

where $\bar{\pi}$ is the 'average per-firm profit upon convergence'.[4] Note that this performance metric assumes a symmetric demand function. Note also that the Nash and monopoly prices p_i^N and p_i^M are not necessarily included in the feasible price sets, so that $\Delta \in \{0, 1\}$ is not necessarily attainable by any policy. The metric Δ is not defined in case a simulation does not converge, but this rarely occurs in our numerical experiments. Where appropriate, we have reported the fraction of simulations that did not converge.

2.4 Main findings by Calvano et. al. (2020)

Calvano et al. (2020*a*) report convergence for 'nearly all' (p. 3276) simulations, with a simulation average of the performance metric Δ that 'ranges from 70 percent to 90 percent' (p. 3277). The authors interpret this as a 'profit gain' and a 'sizable extra-profit compared to the static Nash equilibrium', that is an attractive feature for the firms using the algorithms. Frequencies of different modes of convergence and corresponding averages of Δ are reported in their Table 1. It is concluded that there is often (near-)convergence to strategy equilibria, which they refer to as Nash equilibria (p. 3278), and that 'the algorithms consistently learn to charge supracompetitive prices, without communicating with one another' (p. 3267). The authors are careful, however, not to interpret the supracompetitive limit prices as per se proof of collusion. They write: '[C]ollusion is not simply a synonym of high prices but crucially involves "a reward-punishment scheme designed to provide the incentive for firms to consistently price above the competitive level" (Harrington 2018, p.336). The reward-punishment scheme ensures that the supracompetitive outcomes may be obtained *in equilibrium* and do not result from a failure to optimize' (p. 3269, emphasis in original).

In Section IV.B, Calvano et al. (2020*a*) investigate whether the limit strategies display reward-punishment-like behavior. Upon convergence, one firm is forced to 'defect' in one time period from the limit strategy by applying a 'price cut'. The non-deviating firm responds by lowering its price, and after a number of time periods (typically five to seven, p. 3283), prices return to initial prices. Based on numerical results reported in their Figure 4 and Table 2 and 3, it is claimed that: '[c]learly, the deviation is punished' (p. 3282) and '[o]ur algorithms [...] consistently learn to restart cooperation after a deviation.' (p. 3283). Because of the effect of this response on the profit of the deviating firm, this pattern is interpreted as a punishment mechanism. The authors conclude that '[c]ollusion [...] is enforced by punishment in case of deviation' (p. 3295) and 'do not result from a failure to optimize' (p. 3269) - contrasting their work with Waltman and Kaymak (2015) which 'may not be collusion but a failure to learn' (p. 3269), and Cooper, Homem-de Mello and Kleywegt (2015) which is referred to as 'collusion by mistake' (p.

[4]From the description in Calvano et al. (2020*a*) (p. 3277) we interpret $\bar{\pi}$ to be the *observed* average profit over the time periods $t_0 + 1, \ldots, t_0 + 100,000$.

3269).

These various ex post verifications lead the authors to assert that supracompetitive prices generated by their Q-learning algorithm 'are sustained by collusive strategies' (p. 3267) and to conclude that Q-learning pricing algorithms 'systematically learn to collude' (p. 3295). This should 'ring an alarm bell' (p. 3295) with competition authorities, as software programs are 'increasingly being adopted by firms to price their goods and services' (p. 3267), yet 'leave no trace whatever of concerted action: they do not communicate with one another, nor have they been designed or instructed to collude' (p. 3295). While nowhere in the paper do the authors explicitly say that firms may be interested to adopt Q-learning for its collusive limit properties, it is suggested that firms using them would benefit: for example, training them off-line could set them up to 'start to collude the moment they engage in real action' (p. 3293).

In several places, Calvano et al. (2020*a*) acknowledge that the algorithms take a very long time to converge and that this is potentially problematic for all practical purposes. It is stated that 'the number of repetitions required for completing the learning is typically high, on the order of hundreds of thousands' (p. 3269). More specifically, convergence takes between about 400,000 and several millions of time periods (p. 3276). The authors attribute the slow learning to the algorithms updating 'only one cell of the Q-matrix at a time, and approximating the true matrix generally requires that each cell be visited many times' (p. 3272). In Section VI, they write that it is crucial to address the time scale of collusion in future research (p. 3295), but suggest that alternative measures of collusion, off-line training, applications with very short time periods, or alternative algorithms can speed up convergence. Calvano et al. (2020*a*) assert that, despite the slow convergence, the firms obtain 'a sizable extra-profit compared to the static Nash equilibrium' (p. 3277) and conclude that '[a]ny conclusions are necessarily tentative at this stage, but our findings do suggest that algorithmic collusion is more than a remote theoretical possibility' (p. 3268). For practical competition policy purposes, the authors warn: '[A]lgorithmic collusion might not be that improbable' (p. 3295).

3 Dynamics and limits of Q-learning

In this section, we analyze the dynamics of Q-learning in more detail, in order to shed light on why independent Q-learning algorithms sometimes converge to 'collusive' outcomes and why this takes a long time. Section 3.1 shows in a simplified setting (for a constant exploration rate ϵ_t and just two prices) how collusive strategy-equilibria exist by construction of the algorithm, and are only a subset of all possible limiting strategy-equilibria. Section 3.2 explains why potential convergence to collusive strategy pairs is intrinsically slow and cannot be sped up by obvious changes to the algorithm. Section

3.3 reviews the claim that the limiting strategies are collusive.

3.1 Pre-programmed collusive strategy-equilibria

To develop understanding of the dynamics of ϵ_t-greedy Q-learning, we consider the system where both players use Q-learning with *fixed* exploration probability $\epsilon \in [0, 1]$. Let any mapping from $\mathcal{A}_i \times \mathcal{A}_{-i}$ to $\mathcal{A}_i$ be called a *strategy* of player i, and let Σ_i denote the collection of all such mappings. A strategy $\sigma^{(i)} = \{\sigma^{(i)}(s) : s \in \mathcal{A}_i \times \mathcal{A}_{-i}\}$ of player i is called (δ, ϵ)-*best-response* to strategy $\sigma^{(-i)} = \{\sigma^{(-i)}(s) : s \in \mathcal{A}_{-i} \times \mathcal{A}_i\}$ of player $-i$ if, for all states $s = (s_i, s_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$,

$$\sigma^{(i)}(s) \in \arg\max_{p \in \mathcal{A}_i} \frac{\epsilon}{|\mathcal{A}_{-i}|} \sum_{p_{-i} \in \mathcal{A}_{-i}} \{\pi_i(p, p_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(p, p_{-i})\} + (1-\epsilon)\{\pi_i(p, \sigma^{(-i)}(s_{-i}, s_i)) + \delta V^{(i)}_{\sigma^{(-i)}}(p, \sigma^{(-i)}(s_{-i}, s_i))\}, \tag{3}$$

where, for all states $s = (s_i, s_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$,

$$V^{(i)}_{\sigma^{(-i)}}(s) := \max_{p \in \mathcal{A}_i} \frac{\epsilon}{|\mathcal{A}_{-i}|} \sum_{p_{-i} \in \mathcal{A}_{-i}} \{\pi_i(p, p_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(p, p_{-i})\} + (1-\epsilon)\{\pi_i(p, \sigma^{(-i)}(s_{-i}, s_i)) + \delta V^{(i)}_{\sigma^{(-i)}}(p, \sigma^{(-i)}(s_{-i}, s_i))\}. \tag{4}$$

Equation (4) defines the value function or optimal-value-to-go function $V^{(i)}_{\sigma^{-i}}(\cdot)$ for player i when playing against strategy $\sigma^{(-i)}$, or, more precisely, when the opponent plays according to $\sigma^{(-i)}$ with probability $1-\epsilon$ and selects an action uniformly at random with probability ϵ. Equation (3) defines the corresponding optimal actions. Note that these value functions incorporate the random exploration of the competing firm, but not the firm's own random explorations. This reflects the fact that Q-values are updated after determining the price and not before.[5] A strategy pair $(\sigma^{(i)}, \sigma^{(-i)}) \in \Sigma_i \times \Sigma_{-i}$ is called a (δ, ϵ)-*strategy-equilibrium* if they are mutually (δ, ϵ)-best-response to each other.

To get insight in the potential existence and structure of (δ, ϵ)-equilibria that can be called 'collusive' and how and when they may be learned, we focus further on the setting with $m = 2$ feasible prices for each firm: the Nash price p^N and the monopoly price p^M. Since $m = 2$ corresponds to only 256 possible strategy pairs, this allows us to compute all (δ, ϵ)-strategy-equilibria by brute force enumeration, for any given feasible values of δ and ϵ.[6] In usual notation for the prisoner's dilemma, write $C := p^M = 1.9250$ for the action

[5] More precisely, Q-learning is an off-policy reinforcement learning algorithm, i.e., it calculates an optimal policy assuming no own exploration, but behaves according to, in this case, an ϵ_t-greedy policy based on the optimal policy. The optimal ϵ_t-greedy policy may, however, differ from the ϵ_t-greedy policy derived from the optimal policy (see Van Seijen et al., 2009).

[6] For general m there are m^{2m^2} strategy pairs, which for $m = 3$ already equals 387,420,489, and for $m = 15$, which Calvano et al. (2020*a*) analyze, is a number with 530 digits.

'comply', $D := p^N = 1.4729$ for the action 'defect', and write $R := 0.3375$, $S := 0.1180$, $T := 0.3679$, $P := 0.2229$ for the corresponding payoffs in the baseline parametrization. The single-period pricing game is now equivalent in structure to a prisoner's dilemma, with payoffs given in Table 1.

	C	D
C	(R, R)	(S, T)
D	(T, S)	(P, P)

Table 1: Instantaneous payoff $(\pi_i(p_i, p_{-i}), \pi_{-i}(p_i, p_{-i}))$ for player i (row) and $-i$ (column).

There are four possible states, two feasible actions, and therefore $2^4 = 16$ possible strategies $\{CCCC, CCCD, CCDC, CCDD, \ldots, DDDD\}$, where the four consecutive letters of a strategy refer to the actions taken in state (C, C), (C, D), (D, C), and (D, D), respectively, in that order. Thus, for example, $DCDD$ is the strategy that plays D in state (C, C), C in state (C, D), D in state (D, C), and D in state (D, D). For each strategy $\sigma^{(-i)} \in \Sigma_{-i}$ and for each $\delta \in (0, 1)$ and $\epsilon \in [0, 1]$, we have computed all possible (δ, ϵ)-strategy-equilibria by self-consistently solving the Bellman equations (3) and (4).[7] These are reported in Table 2 and visualized in Figure 1.

These computations reveal that there are one to three strategy-equilibria, depending on the values of δ and ϵ, which all turn out to be symmetric: both players using 'All Defect' (AD, which always plays D), 'Grim Trigger' (GT, which plays C in state (C, C), and D otherwise), or 'Win-Stay-Lose-Shift' (WSLS, which plays C in states (C, C) and (D, D), and D otherwise). The strategy AD consists of always playing the Nash price. The AD equilibrium always exists, but is clearly not collusive. The other two strategies, WSLS and GT, can be interpreted as having a 'reward-punishment scheme', as they play C in state (C, C) ('rewarding' the competitor playing C) and play D in state (C, D) ('punishing' the competitor who deviates from C). GT equilibria exist for the largest set of (δ, ϵ)-values in Figure 1. Note that supracompetitive prices generated by GT are unstable: a single D, caused, e.g., by random exploration, will cause both players to shift to playing D forever. Whether GT can be called collusive thus depends on one's definition: if collusion only means having a reward-punishment scheme then GT clearly qualifies, but if one also requires that collusion does not break down forever after a single defect, then WSLS is the only candidate.

Convergence to collusive equilibria is thus possible because there exist collusive (δ, ϵ)-strategy-equilibria that solves the Bellman equations (3) and (4); letting ϵ_t slowly decrease from one to zero allows the system to converge, on some sample paths, to such an equilibrium. For the case of $m = 2$ there are three such collusive equilibria. WSLS exists

[7] For details of these computations see Barfuss and Meylahn (2022) and Meylahn and Janssen (2022).

Strategy of both players	Conditions for equilibrium
DDDD ('All Defect')	No additional conditions
CDDD ('Grim Trigger')	$\frac{\epsilon(S+T-R-P)+2(R-T)}{(\epsilon^2-3\epsilon+2)(P-T)} < \delta < \frac{\epsilon(S+T-R-P)+2(P-S)}{(1-\epsilon)(\epsilon(T-P)+2(P-S))}$
CDDC ('Win-Stay-Lose-Shift')	$\delta > \frac{\epsilon(P+R-S-T)+2(T-R)}{(1-\epsilon)(\epsilon(P+S-T-R)+2(R-P))}$

Table 2: Conditions on $\delta \in (0,1)$ and $\epsilon \in (0,1)$ such that the strategies in the left column form (δ, ϵ)-strategy-equilibria if used by both players. For $\epsilon = 1$ and any $\delta \in (0,1)$, All Defect is the only equilibrium.

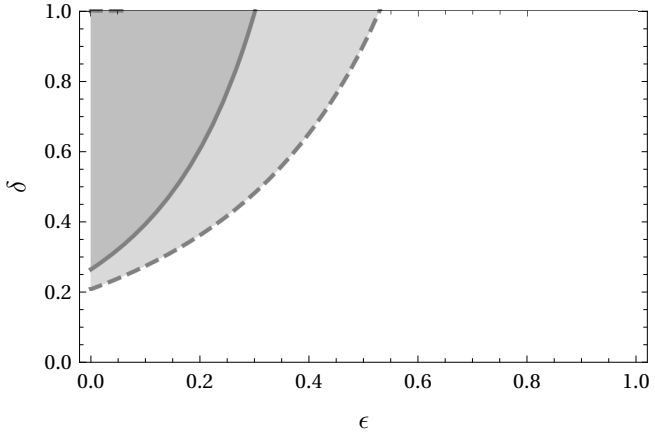

Figure 1: Phase diagram of (δ, ϵ)-strategy-equilibria. AD is always an equilibrium, GT is an equilibrium if (δ, ϵ) lies above the large dashed curve and below the tiny dashed curve in the upper left corner (light and dark gray areas), and WSLS is an equilibrium if (δ, ϵ) lies above the solid curve (dark gray area).

for high enough δ and low enough ϵ, i.e., such that the WSLS condition in Table 2 is satisfied. This condition corresponds to the points (ϵ, δ) lying above the solid line in Figure 1. Similarly, GT exists when the GT condition in Table 2 is satisfied, corresponding to the points (ϵ, δ) lying between the two dashed curves (the upper dashed curve is barely visible in the top left corner) in Figure 1. And, finally, AD exists everywhere in Figure 1. Figure 1 also reveals that WSLS or GT can only be collusive equilibria if ϵ is sufficiently small.

For Calvano et al. (2020*a*)'s case with $m = 15$, there are similar types of collusive equilibrium strategies, yet explicitly computing and characterizing them, as we can do for $m = 2$ in Table 2, is not tractable by brute-force computation. It is not clear how many equilibria there are, and how to characterize or categorize them into collusive or non-collusive. What is the same for $m = 2$, $m = 15$, or any number of prices, is that the existence of and possible convergence to collusive strategy-equilibria is pre-programmed, as it were, in the dynamical system, as a consequence of the choice of the state space, the action space, and the decreasing exploration rates — in addition, obviously, to the assumption that both players use this same algorithm to begin with.

3.2 The Q-learning algorithm is intrinsically slow

The speed with which the algorithms converge depends on the decreasing exploration rate in two ways: (1) there is an initial period in which the collusive strategies do not exist yet, and (2) after they exist a specific sequence of events has to occur in order to transition to a collusive strategy. To see that collusive equilibria can only be learned after a sufficiently long exploration period because only then do they come into existence, consider first the extreme case of a fixed value $\epsilon = 1$, for which we know there are no collusive equilibria — compare Figure 1. In that case, all immediate payoffs in (4) are state-independent, from which it follows that the value function is state-independent: $V^{(i)}_{\sigma^{(-i)}}(s) = (1-\delta)^{-1} \max_{p\in\mathcal{A}_i} |\mathcal{A}_{-i}|^{-1} \sum_{p_{-i}\in\mathcal{A}_{-i}} \pi_i(p, p_{-i})$ for all states s. In the baseline parametrization, there is a unique price $p^* = 1.58271$ at which the optimum in this maximization problem is attained, and the strategy σ^* defined by $\sigma^*(s) = p^*$, for all states s, constitutes a $(\delta, 1)$-strategy-equilibrium (σ^*, σ^*). Because this strategy always sets the optimal price assuming that the opponent plays uniformly at random, and therefore does not contain any structure that can be interpreted as 'reward' or 'punishment', this strategy can be considered 'competitive', i.e., non-collusive, and the equilibrium (σ^*, σ^*) can be considered a non-collusive equilibrium of strategies.[8] By continuity properties of $V^{(i)}_{\sigma^{(-i)}}(s)$, considered as a function of ϵ, it follows that this equilibrium is the only one possible if ϵ is not too small. This is formalized in the following theorem, which we prove in Appendix C.

Theorem 1 *For all $\delta \in (0,1)$ there is an $\epsilon(\delta) \in [0,1)$ such that, for all $\epsilon \in (\epsilon(\delta), 1]$, the only (δ, ϵ)-strategy-equilibrium is the non-collusive equilibrium (σ^*, σ^*).*

Q-learning with time-dependent exploration probability $\epsilon_t = \exp(-\beta t)$ does not admit stationary equilibria in the sense as defined above for a fixed ϵ. With ϵ decreasing to zero, the strategy an algorithm tries to learn, which at each point in time depends on ϵ_t, is a moving target. The underlying dynamical system can nevertheless be considered as a system for which the equilibria present at time t are precisely the (δ, ϵ_t)-strategy-equilibria. So considered, Theorem 1 implies that, as long as $\epsilon_t > \epsilon(\delta)$, there is no collusive equilibrium in the system. Hence, convergence to a collusive strategy-equilibrium is not possible before the critical time period $T_{\epsilon(\delta),\beta} := \lfloor -\log(\epsilon(\delta))/\beta \rfloor$ that guarantees that ϵ_t is below the smallest possible $\epsilon(\delta)$ in Theorem 1. If β is very small, as in Calvano et al. (2020*a*), this critical time period will be large, which explains, in part, why the authors find very slow convergence: convergence to a collusive strategy-equilibrium *can* only start to happen after a very long time, when ϵ has declined enough for collusive strategy-equilibria to exist.

[8] This does not mean that Δ will be small; but as explained in Section 4.2, this is simply a consequence of how the feasible price set is constructed.

To validate this reasoning and see whether our characterization of (δ, ϵ)-strategy-equilibria in Section 3.1 for Q-learning with fixed exploration rates ϵ is relevant also in the setting with time-dependent exploration rates $\epsilon_t = \exp(-\beta t)$, we simulate two independent Q-learning algorithms with $\delta = 0.95$ and $\alpha = 0.15$, that is, conform the representative experiment in Calvano et al. (2020*a*, p. 3280).[9] We choose $\beta = 10^{-4}$, which is lower than the representative value 4×10^{-6}, because with two prices less exploration is needed than with fifteen prices.[10] We use the logit demand model with parameters $a_i = 2, c_i = 1$ and $\mu = 1/4$, actions determined as in Calvano et al. (2020*a*) with $\xi = 0.0$, and initial Q-values are chosen as in Calvano et al. (2020*a*), i.e., set at the value of an action given that the opponent prices randomly. It follows from Table 2 that GT is an equilibrium strategy for all $\epsilon < 0.515126$, and WSLS is an equilibrium strategy for all $\epsilon < 0.292042$. In terms of time periods, this means that for all $t < 6633$ only AD is an equilibrium strategy, for $6633 < t < 12308$ both AD and GT are equilibrium strategies, and for $t > 12308$ all three of AD, GT, and WSLS are equilibrium strategies. Based on these computations, we expect that if both players use ϵ_t-greedy Q-learning with $\epsilon_t = \exp(-\beta t)$, strategies will not converge to GT before $t = 6633$ and will not converge to WSLS before $t = 12308$.

These expectations are confirmed by the outcomes of our numerical experiments. The left panel of Figure 2 shows, based on 1000 simulations, how many trajectories are playing the strategy pair AD, GT, WSLS, or something else, as a function of time. Before time $t = 6633$ (the first vertical dashed line) the fraction of trajectories in the AD strategy pair is steadily increasing (after an initial drop caused by the initialization). At time $t = 6633$ the fraction of GT strategy pairs slowly starts increasing, and only at time $t = 12308$ (the second vertical dashed line) WSLS starts to attract strategy pairs – before that, the fraction of sample paths in which both firms play WSLS is negligible. Thus, we see that the structure of (δ, ϵ) strategy-equilibria as defined by the Bellman equations (3) and (4), combined with the fact that ϵ_t decreases from one to zero, gives rise to a *phase transition*.

We further note that from $t = 12308$ to $t \approx 40000$, in the left panel of Figure 2, 41 percent of the trajectories start converging to WSLS. The other equilibria eventually attract 3.6 percent (AD) and 2.4 percent (GT) of the samples. Hence, about 53 percent of the simulations converge to limiting strategies pairs that may have supracompetitive prices but are not strategy equilibria. The right panel of Figure 2 shows the fraction of trajectories that a *single player* spends on AD, GT, WSLS, and other strategies, as a function of time. In this case, the fractions converge to 4.8 percent for AD, 4.1 percent for GT, 43 percent for WSLS, and 48.1 percent for other, non-equilibrium strategies. We thus observe that the majority of limit equilibria and limit strategies are not collusive.[11]

[9] Table 5 in Appendix A shows that this choice of α has the right order of magnitude.

[10] Table 3 shows that alternative choices lead to qualitatively similar findings.

[11] It is worth emphasizing that these findings are conceptually different from the observation reported in Table 1 in Calvano et al. (2020*a*) that only about one half of their simulations converged to 'Nash

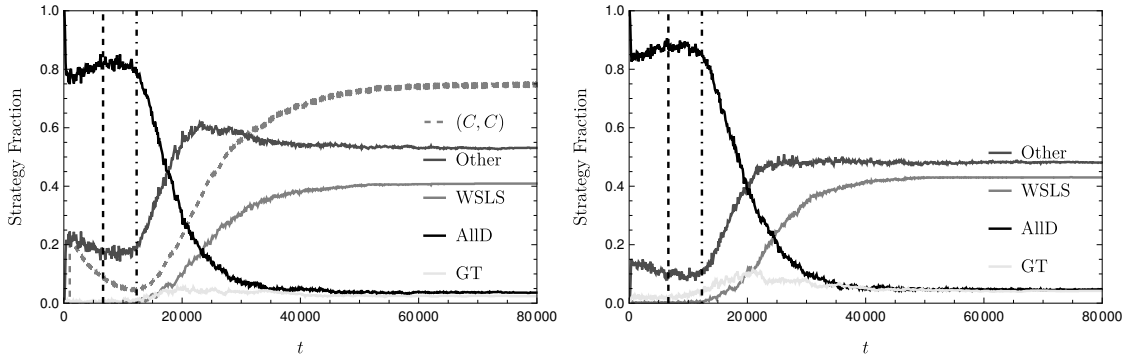

Figure 2: Left: Fraction of trajectories (based on 1000 simulations) in the AD strategy pair (black), GT strategy pair (Light Gray), WSLS strategy pair (Gray) and other strategy pairs (Dark Gray). We also plot the fraction of periods spent in the (C, C) state (Gray dashed) in the 1000 most recent time periods. The horizontal dashed line indicates the time at which the GT strategy pair becomes stable, and the horizontal dash-dotted line indicates the point at which the WSLS strategy pair becomes stable. Right: A simulation with the same parameters, but plotting the strategy fractions for player one only.

The convergence is remarkably slow for a number of reasons, all related to the inner dynamics of ϵ_t-greedy Q-learning. To understand these, consider the strategy-equilibrium WSLS. In order to reach this strategy-equilibrium, it is necessary that action C becomes the optimal action in state (C, C) for both players, i.e., that $Q^{(i)}_{(C,C),C}(t) > Q^{(i)}_{(C,C),D}(t)$ for both $i = \pm 1$. Before $t = 12308$, the majority of optimal strategies according to the Bellman equation (3) are AD, so that, in these cases, $\Delta Q^{(i)}_{(C,C)}(t) := Q^{(i)}_{(C,C),C}(t) - Q^{(i)}_{(C,C),D}(t) < 0$ and state (C, C) can only be reached if both players simultaneously explore and select action C – this happens with probability $\epsilon_t^2/4$, which ranges from 0.02132 at time 12308 to just 0.00008387 at time 40000. Being in state (C, C), $\Delta Q^{(i)}_{(C,C)}(t)$ can increase if both players play C (so that $Q^{(i)}_{(C,C),C}(t)$ is increased after updating) – this initially only can happen by random exploration, i.e., with probability $\epsilon_t^2/4$ – or if both players play D (and $Q^{(i)}_{(C,C),D}(t)$ is decreased after updating). However, in state (C, C), it might also happen that $\Delta Q^{(i)}_{(C,C)}(t)$ *decreases*: namely if player i plays C (e.g., by random exploration) and player $-i$ plays D (so that player i receives the S payoff and $Q^{(i)}_{(C,C),C}(t)$ is decreased after updating), or, conversely, if player $-i$ plays C and player i plays D (so that player i receives the T payoff and $Q^{(i)}_{(C,C),D}(t)$ is increased after updating). Thus, it might take several attempts of reaching state (C, C) by simultaneous random exploration before $\Delta Q^{(i)}_{(C,C)}(t)$ becomes positive.

An additional challenge is that the moments that this happens for both players should not be too far apart: if $\Delta Q^{(i)}_{(C,C)}(t) > 0$ but $\Delta Q^{(-i)}_{(C,C)}(t) < 0$, then, if both players do

equilibria'. Their Table 1 is about (near-)convergence to strategy-equilibria, but does not report how often these limiting strategy-equilibria were collusive – that is, contained a reward-punishment scheme. In contrast, our Figure 2 reveals that in our case more than half of the limiting states are not even collusive equilibria.

not explore (which gets increasingly more likely as time increases), player i will receive the payoff S which will decrease $Q^{(i)}_{(C,C)}(t)$ and might make $\Delta Q^{(i)}_{(C,C)}(t)$ negative again. In fact, even if $\Delta Q^{(i)}_{(C,C)}(t) > 0$ for both players $i = \pm 1$, a series of random explorations in which the players do not play the same action in state (C, C) can again reverse the sign of $\Delta Q^{(i)}_{(C,C)}(t)$ for one of the players. This explains why it may take a considerable time to converge to the WSLS strategy pair, and why, when ϵ_t has gotten too small, it becomes increasingly unlikely that strategy pairs different from WSLS will converge to WSLS.

Hence, we identify two causes why convergence to collusive strategy pairs takes a large number of periods. First, collusive strategy-equilibria only exist, and thus convergence to them is only possible, if ϵ_t is sufficiently small and δ sufficiently large. Because β is very small, it takes a very large number of time periods before the required phase transition occurs. Second, if collusive strategy-equilibria exist — i.e., ϵ_t and δ satisfy the required conditions — it requires a certain number of random events to happen within a particular time frame to establish convergence, which can take a considerable amount of time to materialize. We have made these causes explicit for the case of $m = 2$, but they carry over to any number of prices. In Calvano et al. (2020*a*)'s case of $m = 15$, β is also very small (smaller than 2×10^{-5}), and the dynamics of ϵ_t-greedy Q-learning still requires a number of random events to occur within the right time frame to establish convergence to collusive strategy-equilibria once they exist. Because of these two reasons, slow convergence to collusive strategy equilibria appears to be an intrinsic property of how this algorithm works. It is caused by how the algorithm utilizes the statistical information contained in reward observations – and so not by, e.g., limited computing power or inefficient implementation.

Several ad-hoc changes to the algorithm that present themselves to speed up convergence do not sufficiently do so while also maintaining the same amount of collusion. We consider three natural variations in the setting with $m = 2$ prices and $\delta = 0.95$. In variant (i), we set $\epsilon_t = \epsilon_0 \exp(-\beta t)$ with $\epsilon_0 := 0.292042$ so that convergence to WSLS is immediately possible instead of only at $t = 12308$; the results are presented in Figure 3. In variation (ii), presented in Figure 4, we increased β such that the time $\lceil -\log(\epsilon_0)/\beta \rceil$ until ϵ_0 is reached is much shorter. In variant (iii), presented in Figure 5, the exploration rate was changed to a *fixed* value of ϵ below the critical value ϵ_0. Each of these alternative approaches has drawbacks. Figure 3 shows that variant (i), despite conveniently scaling ϵ_t, still suffers from slow convergence to WSLS, while the fraction of samples converging to WSLS (around 36 percent) is actually smaller than the 41 percent in the baseline. It is moreover not obvious how to choose ϵ_0 in practice, as it depends on the unknown payoff matrix. This problem can potentially be solved by computing a conservative lower bound $\hat{\epsilon}_0 \leq \epsilon_0$ and using $\epsilon_t = \hat{\epsilon}_0 \cdot \exp(-\beta t)$, but that in turn creates the risk of inducing too little exploration. Figure 4 and Table 3 indicate that in variant (ii) increasing β reduces

the amount of collusion, and decreasing β substantially increases the time of the phase transition. Finally, Figure 5 shows that variant (iii), setting a fixed exploration rate, can lead to substantially lower levels of collusion.

We conclude that it is an intrinsic property of this ϵ_t-greedy Q-learning algorithm that convergence to collusive strategy-equilibria, when both players use the algorithm, takes a large number of periods — if it happens at all. This is caused by the structure of the algorithm that creates a critical time before which collusive strategy equilibria do not exist, and that requires a number of random events to happen in a particular time frame to obtain convergence to collusive strategy equilibria. There is no obvious 'quick fix' to ensure that these algorithms converge to collusive strategy-equilibria faster and obvious changes to the hyper-parameters attempting to increase the time of the phase transition decrease the amount of collusion. Our results suggest that algorithmic collusion via these Q-learning algorithms is, literally, a very remote possibility.

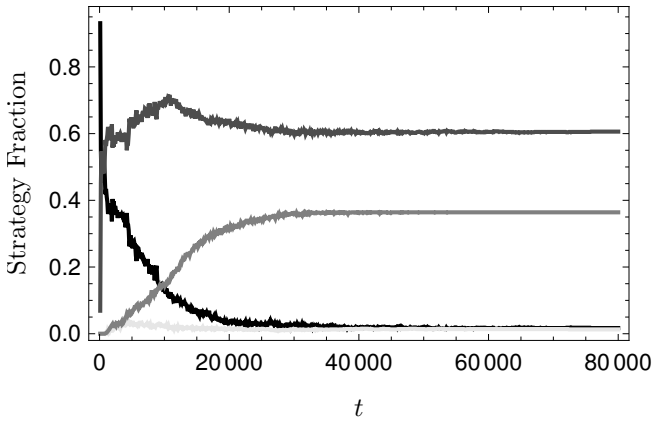

Figure 3: Fraction (1000 samples) of trajectories in the AD strategy pair (black), GT strategy pair (Light Gray), WSLS strategy pair (Gray) and other strategy pairs (Dark Gray). The dashed line indicates the point at which the GT strategy becomes stable, and the dash-dotted line indicates the point at which the WSLS strategy pair becomes stable. We have used $\epsilon(t) = \epsilon_0 \mathrm{e}^{-\beta t}$ with $\beta = 0.0001$ and $\epsilon_0 = 0.292042$.

β	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}
WSLS+GT	0.55 ± 0.03	0.54 ± 0.03	0.46 ± 0.03	0.16 ± 0.02	0.01 ± 0.01
$T_{\epsilon_0,\beta}$	1230860	123086	12309	1231	123

Table 3: The first row gives the sample average and Wilson score interval for the fraction of trajectories converging to the GT or WSLS strategy pairs, based on 1000 trajectories, for different values of β, in the same setting as Figure 4. All trajectories converged. The second row gives the time at which the WSLS strategy becomes stable.

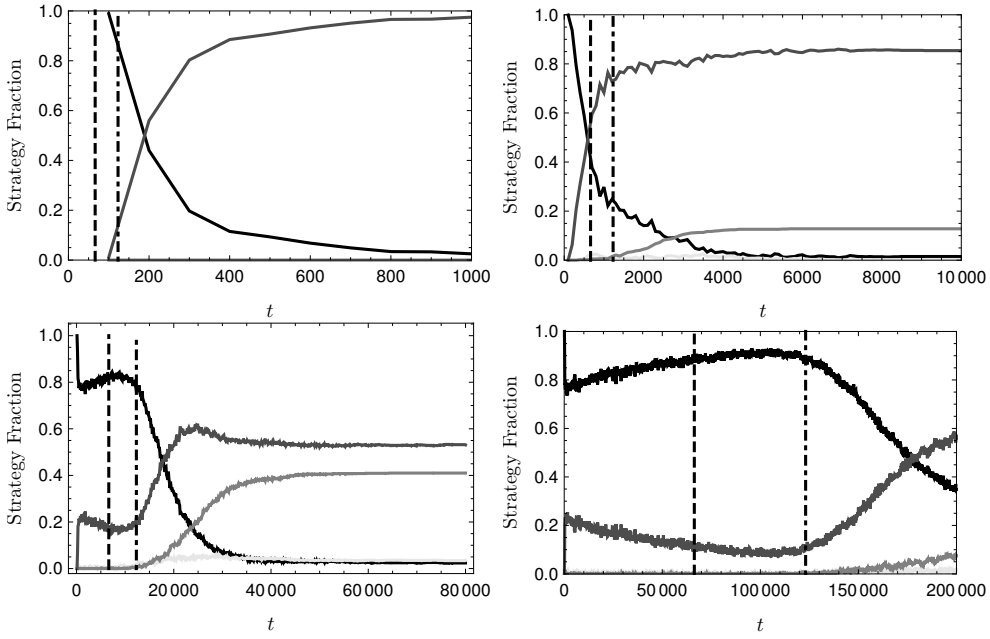

Figure 4: Fraction (1000 samples) of trajectories in the AD strategy pair (black), GT strategy pair (Light Gray), WSLS strategy pair (Gray) and other strategy pairs (Dark Gray). The dashed line indicates the point at which the GT strategy becomes stable, and the dash-dotted line indicates the point at which the WSLS strategy pair becomes stable. We have used $\epsilon(t) = \mathrm{e}^{-\beta t}$ with $\beta = 10^{-2}$ (top left), $\beta = 10^{-3}$ (top right), $\beta = 10^{-4}$ (bottom left), and $\beta = 10^{-5}$ (bottom right).

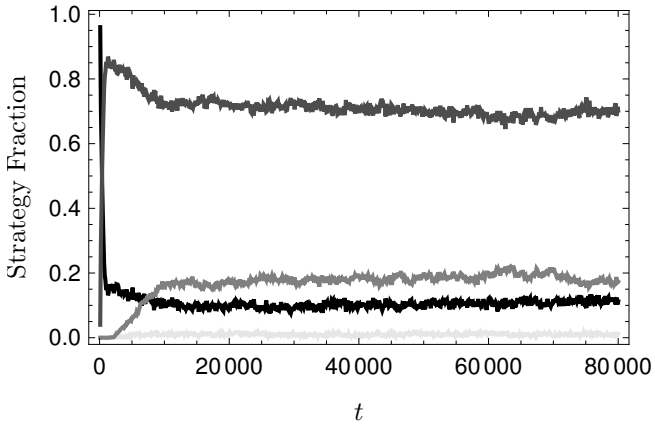

Figure 5: Fraction (1000 samples) of trajectories in the AD strategy pair (black), GT strategy pair (Light Gray), WSLS strategy pair (Gray) and other strategy pairs (Dark Gray). We have used a constant $\epsilon(t) = 0.1$.

3.3 What looks like collusion need not be collusion

In our setting with $m = 2$, we can explicitly identify the nature of the equilibria: there are three equilibrium-types, both firms playing AD, GT, or WSLS, only two of which, GT and WSLS, can be interpreted as having a reward-punishment scheme - where the reward is playing C in state (C, C), and the punishment playing D in state (C, D). Because Calvano et al. (2020*a*) do not consider a strategy collusive if it does not include

a reward-punishment scheme, it follows that, according to their definition of collusion, the strategy pairs GT, GT and WSLS, WSLS are the only collusive strategy-equilibria.[12]

As shown in the left panel of Figure 2, about 57 percent of the simulations do *not* converge to a collusive strategy equilibrium (3.6 percent AD, 53 percent other). If we look at the strategies played by a single firm, then the right panel of Figure 2 shows that about 53 percent of the simulations do *not* converge to collusive strategies (4.8 percent AD, 48.1 percent other). This means that learning is incomplete in the sense that there is no convergence with probability one to a strategy equilibrium. More importantly, it shows for $m = 2$ prices that Q-learning algorithms clearly do not 'systematically learn to play collusive strategies' (p. 3268), if 'systematically' has the common interpretation of convergence with probability one.[13] In addition, the left panel of Figure 2 shows that the fraction of time that both players charge supracompetitive prices converges to about 75 percent. Since random exploration has almost vanished at this point, this implies that a substantial fraction of supracompetitive limit prices are not generated by collusive equilibria.

Calvano et al. (2020*a*) do not explicitly characterize (or formally define) collusive equilibria in the case with $m = 15$ prices. Instead, they aim to show the presence of patterns that could be interpreted as reward-punishment schemes in the limiting strategy outcomes of the learning processes, and take those schemes as an indication of collusive equilibria: '[t]he reward-punishment scheme ensures that the supracompetitive outcomes may be obtained *in equilibrium* and do not result from a failure to optimize' (p. 3269, emphasis original). Starting from any limiting strategy pair that the algorithms has converged to, Calvano et al. (2020*a*) force one firm to cut its price during a single time period, and then count how often this leads to the following pattern: the forced price cut by one firm is followed by a price decrease by the other firm, after a number of time periods the prices return to their original level, and the total discounted profit of the deviating player is lower than if there would have been no deviation at all. The numerical results from such shocks are reported in their Figure 4 and 5 and Table 2 and 3 in support of the claim that deviations are punished. We argue, however, that these results do not warrant the claim, for several reasons.

To begin with, Figure 4 reports *average* prices over all simulations. This fact is ac-

[12] If one also requires collusion to be robust to single-period defections, then the WSLS pair is the only collusive strategy-equilibrium.

[13] To add another layer of complexity, Rothschild (1974) and the two-player generalization Aoyagi (1998) suggest that learning the optimal strategy with probability one is not necessarily a property of optimal policies when future payoffs are discounted – the 'efforts' required on the short-term to ensure this may not outweigh the resulting benefits of higher discounted payoffs on the long-term. Thus, not only is there no convergence with probability one to collusive strategy-equilibria, but 'systematically learn[ing] to play collusive strategies' might in fact not even be desirable.

knowledged by Calvano et al. (2020*a*), but the authors nevertheless conclude '[c]learly, the deviation gets punished' (p. 3282). But based on this figure one cannot conclude that, for example, the majority of sample paths display a pattern like the average. It is in theory possible that in a substantial fraction of the simulations the prices followed a completely different pattern, which can not reasonably be interpreted as 'punishment'. The same holds for their Figure 5 which displays box-plots of the observed price changes. We don't believe it can be concluded from this figure that a significant portion of the simulations displayed the sought-after price pattern: in theory, all kinds of zigzag price patterns are compatible with these box-plots.

Similarly, their Table 2 reports the average price change by both players, for different values of the pre-shock price and forced deviation price, and their Table 3 reports the average profit gains and counts the relative frequency of deviations being unprofitable. These are all simulation averages: the possibility that, in a substantial number of the underlying simulations, the price cut of the deviating player was *not* followed by a price decrease of the non-deviating player is not excluded. It would be helpful to formally define the concept 'reward-punishment scheme' and subsequently count how many simulations satisfied the definition. In the absence of such a definition, it is difficult to assess if their interpretation 'deviation gets punished' is appropriate or that other interpretations such as 'Nash-reversion', 'price war' or 'stick-and-carrot' equally make sense. In addition, for a solid statement on the presence of reward-punishment schemes we believe that one should not only study the effects of one-period price cuts, but also of permanent price increases and decreases.

Let us for now assume that the behavior depicted in Figure 4 of Calvano et al. (2020*a*) is representative for the large majority of samples. Does observing this pattern imply that the underlying limit strategies are collusive strategy equilibria? The answer is no. To see this, consider again the simple setting with $m = 2$ prices. Suppose that firm 1 plays the strategy $CCDC$ and firm -1 plays $CDCC$. Both these strategies are neither GT nor WSLS, so this pair of strategies is not a collusive strategy equilibrium. Suppose that the pre-shock price at time $\tau = 0$ is C for both players, but firm 1 is forced to defect at time $\tau = 1$. The state of player 1 at time $\tau = 2$ is then (D, C) and therefore player 1 defects in period $\tau = 2$. Similarly, the state of player -1 at time $\tau = 2$ is (C, D) and therefore player -1 defects as well in period $\tau = 2$. Because both firms play C in state (D, D), prices return to their pre-shock price C in time period $\tau = 3$. The total discounted profit obtained by the deviating player during these three time periods is $\delta^0 R + \delta T + \delta^2 P + \delta^3 R$. This is strictly smaller than the total discounted profit without deviation, $R \cdot (\delta^0 + \delta^1 + \delta^2 + \delta^3)$, as long as $\delta > 0.2653$.

We thus have constructed a very simple example in which the price cut of one firm is

followed by a price decrease of the other firm, after a number of time periods the prices return to their original supracompetitive level, and the total discounted profit of the deviating player is lower than if there would have been no deviation at all (provided $\delta > 0.2653$). Yet this pattern is generated by strategies that are not a collusive and do not form a strategy equilibrium. One therefore cannot infer from observing these types of price patterns, not on average but also not in individual runs, that the supracompetitive prices converged to and subsequently departed from are supported by collusive equilibria. Price patterns that, with some imagination, can be interpreted as reward-punishment schemes, simply do not imply that they are generated by a collusive equilibrium.[14] Finally, we note that the claim '[o]ur algorithms [...] consistently learn to restart cooperation after a deviation' (p. 3283) clearly does not hold in the $m = 2$ setting: for example, for all simulations that converge to the GT strategy pair, there will be no cooperation after a defect. Thus, even if this claim would be true for $m = 15$, it is not true in general.

4 Performance of the algorithm

Calvano et al. (2020*a*) make the common assumption that the firms aim to maximize their cumulative expected discounted profit. It is therefore natural to evaluate the performance of the firms' pricing algorithms in terms of cumulative expected discounted profit. In particular, in the context of (algorithmic) collusion, it would be natural to compare the cumulative expected discounted earnings obtained under the pricing algorithms with what would have been earned under the Nash prices and monopoly prices, and measure the (firms' benefits of) collusion for example by

$$\begin{aligned}\tilde{\Delta} &:= \frac{\mathbb{E}\left[\sum_{t=1}^{\infty} \delta^t \pi_i(p_i(t), p_{-i}(t))\right] - \sum_{t=1}^{\infty} \delta^t \pi^N}{\sum_{t=1}^{\infty} \delta^t \pi^M - \sum_{t=1}^{\infty} \delta^t \pi^N} \\ &= \frac{(1-\delta)\mathbb{E}\left[\sum_{t=1}^{\infty} \delta^{t-1} \pi_i(p_i(t), p_{-i}(t))\right] - \pi^N}{\pi^M - \pi^N}\end{aligned}$$

However, for evaluating the profit gain from using the pricing algorithms, Calvano et al. (2020*a*) use a different criterion, Δ defined in Section 2.3 above, which is a normalization of the 'average per-firm profit' compared to the static Nash equilibrium *upon convergence*. It is based on this performance measure that the authors suggest the attractiveness of Q-learning for a good 'profit gain'. In light of our finding that convergence on collusive equilibrium strategies can only take place after many periods, the qualification 'upon convergence' is important for the interpretation and conclusions which one can draw from the experimental outcomes.

[14] It is worth emphasizing that small "Q-loss" (Calvano et al., 2020*a*, p. 3278) at most implies near-convergence to strategy-equilibria. It is not relevant to infer convergence to *collusive* equilibria.

We first note that Δ is not a measure of collusion. Δ attains its highest value of one if both sellers always charge the monopoly price.[15] This strategy, of always charging the monopoly price, however, has no 'reward-punishment scheme' in it. Indeed, it will potentially suffer substantial losses when the competitor defects from the monopoly price to the best response. Because, according to Calvano et al. (2020*a*), collusion should include a reward-punishment scheme, the strategy of always charging the monopoly price does not count as collusion. This means that one cannot infer the existence of collusive behavior from high values of Δ.[16] Instead, if one would define collusion by these Q-learning algorithms as convergence to an equilibrium of collusive strategies as in Section 3 then collusion is appropriately measured by the fraction of sample paths converging to such equilibria of strategies (as depicted, e.g., in our Figure 2).

We also note that Δ is not a measure of extra-profit. High values of Δ are by no means an indication that the firms benefit from using the algorithms in terms of their actual objective function, which is the total present discounted value of the expected future payoff stream over the whole time horizon. Because it takes a long time before any supracompetitive prices from collusion might arise (cf. the high amount of AD strategies in Figure 2 during the first 12,000 time periods), the actual contribution of this profit-gain-compared-to-Nash upon convergence to the firm's objective of cumulative discounted profit may be negligible. In the rest of this section, we consider the present discounted value of any future collusive payoff stream in more detail.

4.1 Collusive gains are negligible

To make more precise that the amount of rounds that the algorithm requires for collusion is 'long' also by Calvano et al. (2020*a*)'s own time standard, we define an *effective time horizon* T_δ which is such that all payoffs obtained at time $t > T_\delta$ contribute less than 0.1 percent to the overall profit, and thus are almost irrelevant from the firm's point of view. Let

$$\pi_{\min} := \min_{i=\pm 1} \min_{p_i \in \mathcal{A}_i, p_{-i} \in \mathcal{A}_{-i}} \pi_i(p_i, p_{-i}),$$

$$\pi_{\max} := \max_{i=\pm 1} \max_{p_i \in \mathcal{A}_i, p_{-i} \in \mathcal{A}_{-i}} \pi_i(p_i, p_{-i}),$$

[15] To be more precise, Δ will be close to one if both sellers charge the price in the feasible price set that is closest to the monopoly price, where 'close' depends on the granularity of the price set. See footnote 21 in Calvano et al. (2020*a*).

[16] In several places in the article, Calvano et al. (2020*a*) seem to interpret Δ as a measure of the 'degree of collusion', e.g., on p. 3293, without distinguishing whether the underlying strategies involve reward-punishment schemes or not.

so that all single-period undiscounted payoffs lie in the interval $[\pi_{\min}, \pi_{\max}]$, and define

$$T_\delta := \left\lceil \frac{\log(\frac{\pi_{\min}}{1000\pi_{\max}})}{\log(\delta)} \right\rceil. \tag{5}$$

This definition ensures that

$$\sum_{t>T_\delta} \delta^t \pi_i(p_i(t), p_{-i}(t)) \leq 0.001 \times \sum_{t=1}^{\infty} \delta^t \pi_i(p_i(t), p_{-i}(t)),$$

for all feasible price sequences $\{(p_1(t), p_{-1}(t) \in \mathcal{A}_1 \times \mathcal{A}_{-1} : t \in \mathbb{N}\}$.

In the base parametrization we have $\pi_{\min} = 0.0911$ and $\pi_{\max} = 0.4270$, so that, for $\delta = 0.95$, we find $T_\delta = 165$.

Thus, from the firm's point of view, all profits obtained in the baseline parametrization after the first 165 time periods are practically irrelevant. Hence, the future in which phase changes start to happen, and certainly once there is convergence, after several hundreds of thousands time periods or more – i.e., between about 400,000 and several millions of time periods – is too far away to matter in present discounted value. Calvano et al. (2020*a*) nevertheless assert 'a *sizable extra-profit* compared to the static Nash equilibrium' (p. 3277, emphasis added). The adjective 'sizable' could potentially lead to confusion, since it suggests a significant increase in the firm's objective. In reality, the reported extra-profits are only obtained under a discount factor δ^t, that with $t \gg T_\delta$ is so heavy that it makes their contribution to the firms' objective negligible.

One possible way of narrowing the gap between the effective time horizon and the time it takes to converge to collusive equilibrium strategies, so that some extra-profits may become real, is to choose δ very close to one, so that T_δ becomes larger. Obviously, since T_δ diverges to infinity as δ goes to one, the effective time horizon can be made as long as needed to catch up with the time margins start to appear. T_δ is just over 400,000 for $\delta = 0.999975$, for example. Moreover, Figure 3 in Calvano et al. (2020*a*) appears to suggest that choosing δ close to one can only be beneficial for the amount of collusion, as measured by Δ. However, this turns out not to be a clear-cut solution, as Calvano et al. (2020*a*) briefly note in their footnote 28. The left panel of Figure 6 reports how Calvano et al. (2020*a*)'s measure Δ depends on δ in their baseline parametrization with $m = 15$. The figure reveals that Δ is decreasing in δ if δ is close to one. That the value of Δ remains relatively high, between 70 and 90 percent, is caused by the choice of the feasible price set. In Calvano et al. (2020*a*), the firms choose their prices mostly between the Nash price and the monopoly price, with just a few extreme values below and above. For the right-hand panel in Figure 6, we designed the price space instead to consist of

fifteen prices *symmetrically* spread around the Nash price.[17] The figure shows an even more substantial reduction in Δ as δ approaches one.

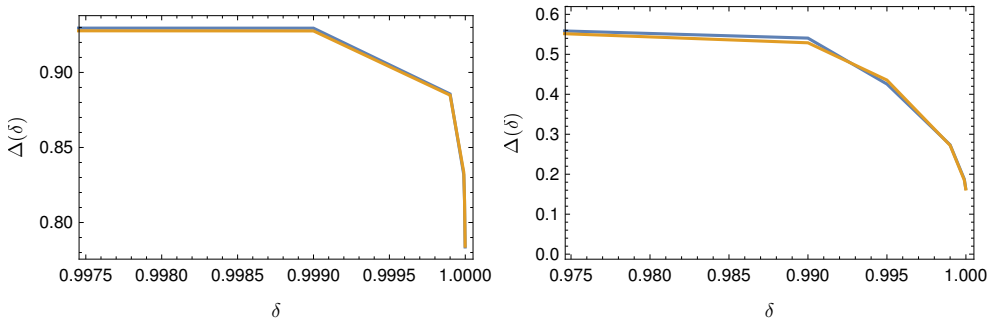

Figure 6: Left: Simulation of 1000 samples for each value of δ. Here we have used the baseline model parameterization with $m = 15$, $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$. We calculate Δ over the last 10^5 time periods and simulate the learners until convergence. All trajectories converged except 2.4% for $\delta = 0.999999$. Right: The same simulation, but with the price interval taken Symmetrically around the Nash price as defined in Appendix B with $\xi = 0.0$. All trajectories converged except 0.3% for $\delta = 0.9999$.

The two left panels of Figure 7 show how the fraction of strategy equilibria depends on δ, in our tractable $m = 2$ setting. The figure reveals that the fraction of collusive equilibria *converges to zero* as δ approaches one. In the right panel of Figure 7 we show how the average time to convergence depends on δ.

A possible explanation for this decrease in collusive equilibrium strategies is that the equilibrium Q-values diverge to infinity as δ approaches one. The Q-values then need to increase significantly during the simulation from their initial values in order to reach their equilibrium values, which appears not to be attainable given the decreasing exploration rate.[18] To remedy this shortage of exploration and give the algorithms more time to converge, a lower value of β could be chosen. The bottom panels of Figure 7 show the result from repeating the simulation, but now with $\beta = 0.00001$. The figure makes clear that decreasing β does not prevent the fraction of collusive equilibria from disappearing when δ approaches one. As expected, on the other hand, it does further increase the time to convergence, as the exploration rate decreases more slowly.

When tuning δ and β, they are in a sense leapfrogging. Increasing δ increases the effective time horizon so that the company may benefit from collusion, but decreases the amount of collusion once δ is sufficiently close to one. Decreasing β counterbalances this by increasing the amount of exploration. While there may be combinations of δ and β for which the company benefits from collusion in time, fine-tuning the parameters to this end makes the claim that the algorithms learn to collude *autonomously* untenable. Note furthermore that the decrease to zero in collusive equilibrium strategies as δ approaches

[17] See Appendix B for the details of this construction.

[18] Simulations indicate that, for example, rescaling the rewards by $(1 - \delta)$ does not solve the issue.

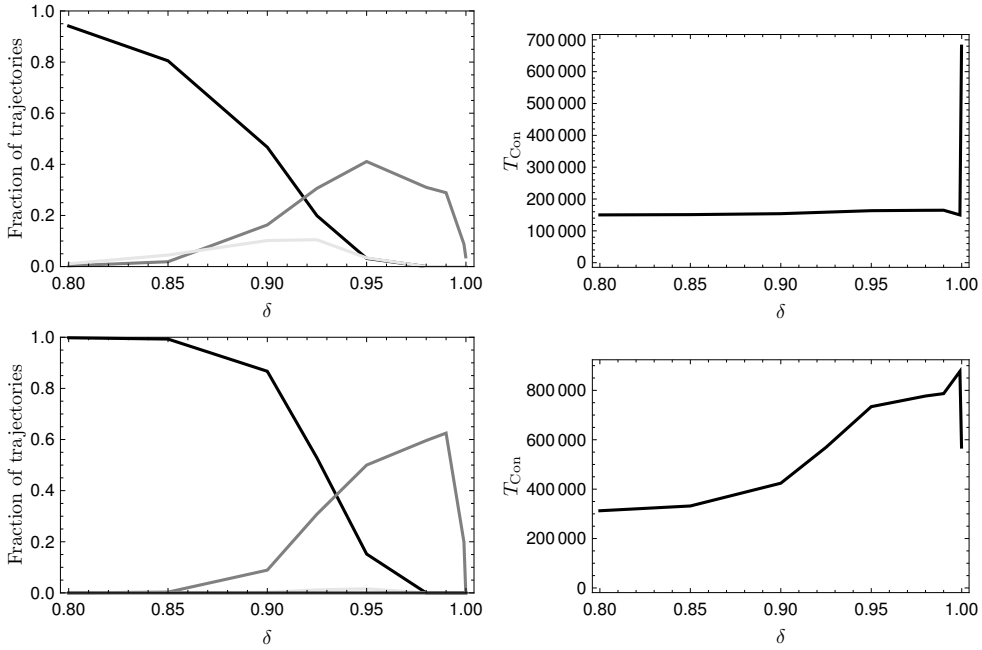

Figure 7: Top Left: Fraction of trajectories (1000 samples per δ) in AD (Black), GT (Light Gray), WSLS (Gray) as a function of δ for the $m = 2$ setting with $\alpha = 0.15$ and $\beta = 0.0001$. Top Right: The average time to convergence for the same simulation as a function of δ. Bottom: Same as top with $\beta = 0.00001$. All trajectories converged.

one does not necessarily translate into a decrease to zero in Δ. The limit strategies may still play the collusive price pair frequently, without being collusive strategies. This again demonstrates the inadequacy of Δ as a measure of collusion with reward-punishment schemes.

4.2 Short-run gains are assumed

The Q-learning algorithms in Calvano et al. (2020*a*) are able to generate positive extra profits within the firms' effective time horizon. However, whether there are short-run gains or losses turns out to depend crucially on the support of the price set. The reason for this is that, for small enough values of δ including $\delta = 0.95$, the Q-learning algorithm is similar to pricing uniformly at random within the effective time horizon. To see this, note that the probability that the Q-learning algorithm does *not* experiment but prices according to the Q-matrices is, within the effective time horizon, bounded from above by $1 - \exp(-\beta_{\max} T_\delta) = 1 - \exp(-2 \times 10^{-5} \times 165) \approx 0.003295$. The corresponding expected number of times that the Q-learning does not experiment is bounded from above by 0.55. Thus, within the effective time horizon of $T_\delta = 165$ time periods, the Q-learning algorithm prices uniformly at random in, on average, more than 164 time periods. The algorithm's actions and payoffs are, in other words, statistically indistinguishable from

pricing uniformly at random.[19]

Any supracompetitive gains to the firms' objective in the short run are therefore attributable to the choice of the feasible price range. In the base parametrization, the average price when both players price uniformly at random is 1.6990, which is substantially larger than the Nash price 1.4729. This translates into average per-period profits of 0.2799, which is substantially larger than the per-period Nash profit of 0.2229. In this sense, any profit gains with a positive present discounted value are caused by the choice of the feasible price sets, which is skewed to be higher than the Nash-price on average. A different feasible price set could just as well result in a worse performance compared to pricing-at-Nash.

An alternative choice, that a priori might seem more natural, is to bound the price set from below by the marginal cost and is symmetric around the Nash equilibrium:

$$\hat{\mathcal{A}}_i = \left\{ c_i + \frac{k}{m+1} \cdot 2(p_i^N - c_i) : k = 1, \ldots, m \right\}. \tag{6}$$

For this support, the average price under pricing-uniformly-at-random is indeed exactly equal to the Nash price, and the average per-period profit when both players price uniformly at random will be 0.17228 or 22 percent *smaller* than the Nash profits.[20] Moreover, for price supports with lower means, the expected profits from adopting the Q-learning algorithm are *lower* than the competitive profits. This goes to show how the choice of the price set determines the average return.[21] Given that without prior knowledge of the demand parameters or without intent to price supracompetitively there is no obvious reason why both firms would choose to learn on price supports that are shifted upwards from the competitive price level, we conclude that the short-run gains are essentially assumed by the choice of the feasible price sets.

[19] This of course changes if δ approaches one (more time periods becoming relevant) or β grows large (less random exploration).

[20] The average price does not determine the average per-period profit when pricing uniformly at random as the average per-period profit depends on the entire payoff matrix. Lower prices generate larger sales. In fact, it is possible to choose an action space with an average price above the Nash price and still have an average per-period profit below the Nash profit when pricing uniformly at random.

[21] Incidentally, changing the parameter ξ, which controls the step between consecutive feasible prices, similarly affects the average price and profit during the effective time horizon. For example, increasing ξ from 0.1 to 1.0 reduces the per-period profit to 0.1916, or 14 percent below the Nash per-period profits in case both firms price uniformly at random around the Nash-price.

5 Q-learning is outperformed by reasonable alternatives

A further critique to the claim that Q-learning algorithms pose a threat of spontaneous collusion is that such algorithmic collusion is not stable against the use of different algorithms. In this section we argue that the equilibrium of strategies which according to Calvano et al. (2020*a*) is 'learned' by an ϵ_t-greedy Q-learning algorithms-with-memory-one is not the appropriate equilibrium concept for algorithmic collusion: the reason is that there are alternative algorithms that easily outperform the Q-learning pricing algorithm. To illustrate this, we consider a simple adaptation of the well-known Exp3 policy (Lattimore and Szepesvári, 2020, Section 11.3), which is easy to implement and comes with a theoretical performance bound. It turns out that Q-learning may face substantial performance losses when playing against Exp3 instead of Q-learning, which implies that Exp3 forms a credible threat to the implementation of Q-learning.

In the basic prisoners' dilemma, the mere existence of mutually advantageous actions does not imply that it is reasonable for rational agents to play these actions, for the obvious reason that there is the threat of the opponent playing defect. Similarly, if it would be the case that mutual use of Q-learning algorithms generate supracompetitive profits, then this does not imply that it is reasonable for the firms to actually use these algorithms. Now, the multi-period pricing game is equivalent to a single-shot game where actions correspond to policies or algorithms that specify, for each possible data set of previously observed prices and (own) payoffs, how the next (possibly random) price should be chosen.[22] Let Π_i denote the policy used by player $i = \pm 1$, let $r_i(\Pi_i, \Pi_{-i})$ denote the corresponding expected discounted total profit (expression (2) with prices generated by the policies), and let $\mathfrak{P}$ denote the space of all feasible policies.[23] For the sake of the argument, suppose that there exists a Nash equilibrium policy pair $(\Pi^N, \Pi^N) \in \mathfrak{P}^2$. Then the mere existence of a policy pair $(\Pi^C, \Pi^C) \in \mathfrak{P}^2$ with $r_i(\Pi^C, \Pi^C) > r_i(\Pi^N, \Pi^N)$ – and possibly with other characteristics, for example that average prices, measured in some appropriate way, are larger than when generated by the Nash policy pair – in itself is not a reason to expect these policies will actually be played by rational agents, for the simple reason that there might be a policy $\Pi^D \in \mathfrak{P}$ that (perhaps significantly) diminishes the payoff when used by the competitor, $r_i(\Pi^C, \Pi^D) < r_i(\Pi^C, \Pi^C)$, and simultaneously increases the payoff of the competitor, $r_{-i}(\Pi^D, \Pi^C) > r_i(\Pi^C, \Pi^C)$.

[22] A policy of player i can formally be defined as a collection of probability mass function $\Pi = \{\Pi(\cdot|h) : h \in H_i\}$, for all possible *histories* in the set of all possible histories $H_i := \cup_{t\in\mathbb{N}}(\mathcal{A}_i \times \mathcal{A}_{-i} \times \mathbb{R})^{t-1}$, such that player i selects price $p \in \mathcal{A}_i$ at stage $t \in \mathbb{N}$, $t \geq 2$ of the game with probability $\Pi(p|(p_i(s), p_{-i}(s), \pi_i(p_i(s), p_{-i}(s))_{1\leq s<t})$, and $\Pi(\cdot|\emptyset)$ denotes the probability mass function of the action at $t = 1$. This definition can be extended to settings with random observations of payoff or demand.

[23] Because in the baseline parametrization both players have the same feasible price set, their corresponding space of feasible policies is also the same.

Now, in this game where actions correspond to policies, it might be intractable to compute equilibrium strategies Π^N; in particular if one takes into account that the demand parameters (a_{-1}, a_1) are *unknown* and, instead of (2), one aims to maximize e.g., the *worst-case* of (2) over all possible demand parameters from a feasible range. Nevertheless, if there is a policy $\Pi^D \in \mathfrak{P}$ that is conceptually simple – meaning that it is not unreasonable to expect that players will consider this policy, for example because its underlying ideas are well-documented in the literature –, not necessarily a Nash policy, but such that $r_i(\Pi^C, \Pi^D) < r_i(\Pi^C, \Pi^C)$ and $r_{-i}(\Pi^D, \Pi^C) > r_i(\Pi^C, \Pi^C)$, then, just as in the prisoner's dilemma, asserting that rational agents might play Π^C requires an explanation why these rational agents do not consider the threat that the competitor plays Π^D. In other words, if there exists a simple price policy that beats Q-learning, then there is work to be done if one wants to argue that there is a real risk that two rational agents independently decide to use Q-learning.

There are many such policies Π^D. One can, for example, design a policy that exploits weaknesses in Q-learning by alternating between AD and WSLS, when WSLS has been learned, in such a proportion that the opponent will not deviate from WSLS. In this section we focus on a simple policy for which one can derive theoretical performance guarantees regardless of the competitor's policy: the well-known Exp3 policy, which we adapt to the setting with discounted rewards. From the perspective of player $i = \pm 1$, this version of Exp3 selects price $p \in \mathcal{A}_i$ in time period $t \in \mathbb{N}$ with probability

$$P_{t,p}^{(i)} := \frac{\exp(\eta \sum_{s=1}^{t-1} \delta^s \hat{X}_{s,p}^{(i)})}{\sum_{p' \in \mathcal{A}_i} \exp(\eta \sum_{s=1}^{t-1} \delta^s \hat{X}_{s,p'}^{(i)})},$$

for all $p \in \mathcal{A}_i$ and all $t \in \mathbb{N}$, where

$$\begin{aligned}
\hat{X}_{t,p}^{(i)} &:= 1 - \hat{Y}_{t,p}^{(i)}, \\
\hat{Y}_{t,p}^{(i)} &:= \frac{Y_t^{(i)} \cdot \mathbf{1}\{p_i(t) = p\}}{P_{t,p}^{(i)}}, \\
Y_t^{(i)} &:= 1 - \pi_i(p_i(t), p_{-i}(t)),
\end{aligned}$$

for all $t \in \mathbb{N}$, $p \in \mathcal{A}_i$ and $i = \pm 1$, and $\eta > 0$ is a hyper-parameter of the algorithm.[24] It is common to measure the performance or *regret* of a policy by comparing it to the best fixed action in hindsight (see, e.g. Lattimore and Szepesvári, 2020, Chapter 11). The

[24]The policy Exp3 can also handle stochastic demand. For example, in the common Bernoulli-logit demand framework in which the random demands $D_{i,t}$ of player i at time t take values in $\{0, 1\}$, we would then define $Y_t^{(i)} = 1 - (p_i(t) - c_i) \cdot D_{i,t}$. Because demand is assumed to be non-random, we replace $p_i(t) \cdot D_{i,t}$ by its expectation.

regret of player i is thus defined as

$$\text{Regret}_i := \max_{p \in \mathcal{A}_i} \sum_{t=1}^{\infty} \delta^t \mathbb{E}[\pi_i(p, p_{-i}(t))] - \sum_{t=1}^{\infty} \delta^t \mathbb{E}[\pi_i(p_i(t), p_{-i}(t))].$$

Suppose that all expected payoffs $\pi_i(p_i, p_{-i})$ are bounded by one – if the unknown parameters (a_{-1}, a_1, μ) are assumed to lie in a bounded set, this can easily be guaranteed by dividing all payoffs by the maximum payoff $\pi_{\max}$ over all feasible unknown parameters. We then have the following theoretical performance bound on Exp3, of which the formal proof is contained in Appendix D.

Theorem 2 *If player i uses Exp3 with hyper-parameter $\eta = \sqrt{\frac{\log(|\mathcal{A}_i|)(1-\delta)^2}{\delta^2 |\mathcal{A}_i|}}$ then, regardless the policy of player $-i$, the regret of player i satisfies*

$$\textit{Regret}_i \leq 2\sqrt{\frac{\delta^2}{1-\delta^2} \log(|\mathcal{A}_i|)}.$$

Table 4 reports what happens when the policies Exp3 and Q-learning play against each other, in the baseline parametrization with (α, β) set to a representative value of $(0.15, 4 \times 10^{-6})$ as in Calvano et al. (2020*a*, p. 3280), and η set to the optimal choice in Theorem 2. We test different discount factors $\delta \in \{0.95, 0.99, 0.999, 0.9999\}$ – note that a discount factor of 0.9999 is not exceedingly high but can be appropriate in practice if, for example, annual interest rate equals 3.7 percent and prices can be updated on a daily basis. Table 4 reports the total expected discounted payoff during the effective time horizon (given by $T_{0.95} = 165$, $T_{0.99} = 842$, $T_{0.999} = 8449$, $T_{0.9999} = 84516$), scaled according to the definition of $\tilde{\Delta}$ to facilitate comparison between different discount factors. Values reported are an average over 1000 samples; we also give 95% confidence intervals.

The results show that, when a firm uses Q-learning, the competitor has incentive to use Exp3 instead. This simultaneously leads to an increased payoff, and a decreased payoff for the Q-learning player. Especially for larger discount values, the loss can be substantial. For lower discount values, the performance of the policies are all roughly the same, indicating that the effective time horizon is too small to facilitate learning.[25] Thus, the policy Exp3 has deep roots in the literature, is easy to implement, comes with a theoretical performance guarantee, and does better than Q-learning against Q-learning. There appears to be no reason to assume that implementation of this policy by a competitor is less likely than implementation of Q-learning.

[25]It is worth mentioning that the positive values of $\tilde{\Delta}$ in Table 4 are a consequence of the feasible price set and do *not* indicate a form of 'collusion'; to see this, Table 7 in the Appendix repeats the experiment of Table 4 for a different price set that is symmetric around the Nash price defined in Appendix B. The results show that Q-learning is then still vulnerable to Exp3, but the absolute payoffs are substantially lower than under the Nash price.

	$\delta = 0.95$		$\delta = 0.99$	
	Exp3	Q-learning	Exp3	Q-learning
Exp3	$.49 \pm .002$	$.50 \pm .006$	$.47 \pm .004$	$.52 \pm .003$
Q-learning	$.48 \pm .003$	$.50 \pm .007$	$.46 \pm .004$	$.50 \pm .003$
	$\delta = 0.999$		$\delta = 0.9999$	
	Exp3	Q-learning	Exp3	Q-learning
Exp3	$.44 \pm .005$	$.56 \pm .002$	$.37 \pm .004$	$.62 \pm .001$
Q-learning	$.38 \pm .005$	$.50 \pm .001$	$.26 \pm .004$	$.49 \pm .0003$

Table 4: Empirically observed $\tilde{\Delta}$ of the row player in the baseline parametrization, for all combinations of Exp3 and Q-learning and different values of δ.

As discussed in Section 3.3, Calvano et al. (2020*a*) state that the presence of a reward-punishment scheme is required for an equilibrium strategy to be collusive – merely observing supra-competitive prices is not enough. But what type of equilibrium is relevant? As explained in Section 3 in the context of a 2-action 1-memory prisoner's dilemma, the Bellman equation (4) induces *equilibria of strategies*, where a strategy is a mapping from states (prices charged in the previous period) to actions (prices charged in the current period). These equilibria of strategies can potentially be 'learned' if both players use a particular variant of Q-learning with the same underlying state space. However, *on the level of algorithms or policies, Q-learning is not in equilibrium.* This is clear from our numerical results above, which show that, if both players use Q-learning and δ is not too small, each player has an incentive (sometimes substantial) to play Exp3 instead.

The concept of a 'strategy', i.e., a mapping from a state space to an action space, can be thought of as a human interpretation of the dynamics generated by a particular algorithm. But algorithms differ in their corresponding spaces of feasible strategies, their underlying state spaces, and in whether 'strategies' are actually an appropriate interpretation at all. For example, Q-learning with memory one may be said to be 'trying to learn' a mapping from all feasible price pairs to all feasible prices, but for Exp3 and other algorithms without memory this does not apply. In its most general form, an algorithm or policy is a mapping from *all available data* to (probability distributions on) the action space. The only true state is 'all available data', and the only true state space is all possible collections of data available at a particular time point. Whether or not the dynamics of algorithms can (approximately) be described, as t grows large, by a mapping that only uses a *subset of the available data* (as in Q-learning) is an algorithm-specific question that does not hold in generality, and that may not even be a desirable property in terms of performance and resilience against rational opponents.

We conclude that 'equilibrium of strategies' is an algorithm-specific concept that is often

incommensurable between different algorithms, and therefore *is not the appropriate equilibrium concept* when comparing potentially colluding algorithms. Instead, we believe that it is more natural to compare algorithms in terms of what they contribute to *the players' actual objective* – in this case the total expected infinite-horizon discounted pay-offs. Table 4 shows that Q-learning is not an algorithm in (or nearly in) equilibrium: there is often a clear incentive to play Exp3 instead of Q-learning – an easy-to-implement alternative with theoretical performance bounds and roots in the scientific literature. Exp3, moreover, is just one instance of a potentially much larger set of algorithms against which Q-learning performs poorly – including, for example, policies that exploit weaknesses in Q-learning. This is difficult to reconcile with the idea that Q-learning would be implemented in practice by rational agents.

Let us emphasize that we do not mean that algorithmic collusion is only convincing if the corresponding algorithms are in equilibrium on the level of algorithms – for one thing, characterizing such equilibria in repeated games with unknown pay-off matrices is often an intractable problem. However, we do believe that any claim that a particular algorithm is 'collusive' should be accompanied by an analysis that (a) either shows that the algorithm would perform well against 'reasonable' competitive alternatives, or (b) argues why in the particular market circumstances it is credible that a player will use the algorithm despite its potentially poor performance against competitive opponents.

6 Concluding remarks

We critically examine claims that firms learn to charge supracompetitive prices that are supported by collusive equilibria, if they use the same ϵ_t-greedy Q-learning algorithm. We focus on Calvano et al. (2020*a*) but our conclusions hold more generally for these types of simple Q-learning algorithms. Section 6.1 summarizes our main findings and Section 6.2 sets out what is needed to demonstrate the existence of a colluding price algorithm that does form a threat to competition.

6.1 Conclusions on the alleged algorithmic collusion

Calvano et al. (2020*a*) claim that their 'results indicate that, indeed, relatively simple pricing algorithms systematically learn to play collusive strategies' (p. 3268). Their concept of collusion (p. 3269) involves that such strategies should include a reward-punishment scheme. We note, however, that their claim of learning to play collusive strategies is based in part on observing high values of the measure Δ. Yet a high value of Δ is not an indication that such schemes are present – a strategy of always pricing supracompetitively, for example, scores very high on Δ but does not count as collusion according to Calvano et al. (2020*a*). In addition, if one would measure collusion by the fraction of

times that the algorithms converge to a strategy-equilibrium that includes some sort of reward-punishment scheme, our analysis for the tractable case of $m = 2$ prices reveals that a substantial number of simulations (more than half) do *not* converge to collusive equilibria or even collusive strategies. Calvano et al. (2020*a*) do not offer evidence or reasons to believe that the $m = 15$ case would be structurally different.

Calvano et al. (2020*a*) claim that the supracompetitive prices generated by their algorithms are 'sustained by collusive strategies' (p. 3267), 'do not result from a failure to optimize' (p. 3269), and that a 'reward-punishment scheme ensures that the supracompetitive outcomes may be obtained in equilibrium' (p. 3269). To support these claims, the authors observe in limiting strategies a particular price pattern – a forced unilateral price decrease is followed by price cuts of the non-deviating player, and after a number of time periods prices gradually return to their original level – which they then interpret as a 'reward-and-punishment scheme'. First note that a solid statement on the presence of reward-punishment schemes should preferably not only be based on one-period price cuts but also on permanent price increases and decreases. Moreover, the price patterns that Calvano et al. (2020*a*) analyze may allow other interpretations. In particular, we show in a simple example that the existence of such price patterns can also be generated by non-collusive non-equilibrium strategies. Hence, the main argument to infer that the supracompetitive prices generated by their Q-learning algorithms are 'sustained by collusive strategies' appears to be incorrect.

In response one might say that being in equilibrium is irrelevant, and that what matters only is the presence of rewards and punishments. However, if firms do not play equilibrium strategies, then the algorithms have not converged to best-response strategies that a rational agent would play. The supra-competitive outcomes would then truly be 'collusion by mistake'. Indeed, our simulations for the tractable version with $m = 2$ prices show that a substantial fraction of supracompetitive prices in the limit are *not* explained by collusive strategies or equilibria. Note that we do not claim that almost all limiting strategies in the $m = 15$ setting are not collusive strategy pairs, but rather that the argument used to infer collusive equilibria is incorrect. It therefore remains possible that a substantial fraction of the supracompetitive outcomes that Calvano et al. (2020*a*) find is not sustained by collusive equilibria.

Moreover, we have explained that the notion of equilibrium underlying Calvano et al. (2020*a*) is not the right concept. If both firms use the same ϵ_t-greedy Q-learning algorithms with memory one, then this creates a type of strategy-equilibria of which some can be called collusive. However, there is no good reason why a competing firm would use exactly this algorithm. The equilibrium-of-strategies is a concept that hinges on both players using the same restrictive type of algorithm with the same restricted state space.

Thus, this is an algorithm-dependent equilibrium, which is not the appropriate equilibrium concept for the meta-game in which firms choose algorithms in order to enhance their profit. In particular, a user of a Q-learning algorithm will not only be interested in the performance against another, identical Q-learner, but also in the performance against reasonable alternative price policies. We show that when the Q-learning algorithm plays against a simple alternative, Exp3, performance can deteriorate substantially, which implies that, on the relevant level of algorithms, Q-learning is far out of equilibrium.

Calvano et al. (2020*a*) claim that their algorithms generate 'sizable extra-profit compared to the static Nash equilibrium' (p. 3277). We believe that this claim is potentially confusing. The measure Δ that is used to quantify this extra profit is namely not based on the actual objective of the firms – the total discounted payoffs over an infinite time horizon – but on profit 'upon convergence' (p. 3277). High values of Δ do not imply a sizable increase in the firms' actual objective. In addition, using the notion of an effective time horizon, it follows that all potential extra-profits 'upon convergence' are in fact negligible for the firms' objective. All extra profits that do contribute to the firms' objective are caused by the expected earnings in case both players price uniformly-at-random. While pricing uniformly-at-random happens to generate supracompetitive prices and payoffs on average for the demand parameters and feasible price set used by Calvano et al. (2020*a*), alternative demand parameters and feasible price sets can easily generate sub-competitive payoffs. In other words, all observed extra-profits in the firms' actual objective can be attributed to factors not related to collusion.

We have also demonstrated that convergence to collusive equilibria in the algorithm studied by Calvano et al. (2020*a*) is intrinsically slow: it requires a phase transition to take place that takes considerable time to realize and before which collusive strategy-equilibria do not exist. In addition, convergence after the phase transition requires certain random events to happen within a particular time frame to establish convergence, which can take a long time and can easily fail. By simulating various ad-hoc changes to the parameters ϵ_t and δ we have shown that obtaining a 'sizable extra-profit' *within* the effective time horizon, i.e., fast enough such that it actually contributes to the firms' objectives, is not possible with this algorithm in the base-line parametrization. Because with $m = 15$ prices it presumably takes considerably longer to learn collusive equilibria, there is reason to believe that the problem of prohibitively slow convergence will not be diminished but rather amplified if there are fifteen instead of two feasible prices. More sophisticated algorithms can perhaps generate supra-competitive gains within the effective time horizon, even for $m = 15$ prices, but that then first needs to be demonstrated. Reducing the time to convergence from more than 400,000 to 165 periods is a more than 2400-fold reduction that might be impossible even for the most sophisticated algorithms.

Our overall conclusion is that the simulations presented by Calvano et al. (2020*a*) do not give sufficient evidence for the claim that these types of Q-learning algorithms systematically learn collusive strategies: that claim is incorrect for $m = 2$ prices and we don't believe that a convincing reason is offered why this would be different for $m = 15$ prices. The same is true for the claim that the supracompetitive prices generated by the algorithms are often supported by collusive equilibria. That the algorithms would generate sizable extra-profit is true for some price-sets and false for other price-sets, and is determined by factors unrelated to collusion. Based on these findings, we conclude that warnings that algorithmic collusion via these types of Q-learning algorithms is 'more than a remote possibility' (p. 3268) or 'not that improbable' (p. 3295) and should 'ring an alarm bell' (p. 3295) with competition authorities, are premature.

In this paper we have focused on the criticisms summarized above. However, there are at least two other aspects of Q-learning that one could scrutinize. First, acknowledging that convergence of their algorithms takes a very long time, Calvano et al. (2020*a*) suggest that offline training can be used to speed up convergence. However, a main challenge in algorithmic pricing is that the unknown market demand parameters have to be learned. These parameters can only be learned by interactions with actual demand. It is far from clear how a firm can learn about its consumers' preferences in offline isolation – unless it is some kind of market research, but this does not seem to be meant. Contrary to AlphaGo or other board games, pricing games crucially involve unknown market parameters.

Calvano et al. (2020*a*) acknowledge that 'the training environment may not exactly reflect the reality of the markets in which the algorithms will be deployed. This implies that what an algorithm has learned offline may be of little help in colluding in real life' (p. 3293). Yet the authors nevertheless infer from a small numerical experiment that 'offline learning may not be completely useless after all' (p. 3294). This conclusion is open to various criticisms. It is based on Δ in their Figure 11, which does not measure collusion or profit gain according to the firms' actual objective, as explained.[26] Moreover, even if it would, the time scale of their Figure 11 is still outside the effective time horizon. Without giving details on an offline-learning plan and how it relates to the market reality in which the firms are subsequently supposed to collude, and without studies that show positive effects on the amount of collusion (properly measured) and the time-to-convergence, the assertion 'offline learning may not be completely useless after all' (p. 3294) is not sufficiently substantiated.

A second issue is that the (identical) pricing algorithms in Calvano et al. (2020*a*) are assumed to start at *exactly* the same moment and use *exactly* the same parameters, such

[26]The parenthetical remark 'The original levels of collusion can be reproduced' on p. 3294 is another example where Calvano et al. (2020*a*) incorrectly associate Δ with collusion.

as the learning parameters α and β, the feasible price ranges $\mathcal{A}_i$ and the initial Q-matrices $Q^{(i)}(0)$. Moreover, the latter two quantities depend on the payoff functions $\pi_i(\cdot,\cdot)$, which are assumed to be *unknown* to the firms – computing the initial Q-matrix (equation (8) on p. 3275) requires knowledge of the *complete* payoff matrix of a firm. The algorithm is thus using information that is not available to the firms employing them. At a minimum, this contradicts the authors' claim that their algorithm has 'no prior knowledge of the environment in which they operate' (p. 3268). It might be possible to fix some of these issues without drastically changing the observed qualitative behavior – although relaxing the assumption of synchronized starting times might not be trivial. We also expect the outcomes to be robust to both firms independently choosing α and β within natural bounds. Nevertheless, the structure and synchronization assumed raises questions about the autonomous nature of collusion that is emphasized as alarming. The current setup of Calvano et al. (2020*a*) at least gives the impression that coordination on starting times, fine-tuned hyper-parameters, and a conveniently elevated feasible price set is required.

6.2 Conditions for algorithmic collusion

Algorithmic collusion clearly is an important topic, that rightly moves antitrust priorities and large amounts of enforcement budgets. Yet there are many open questions still. How will the increased use of data-driven pricing algorithms affect competition in the decades to come? Will it enhance price competition amongst suppliers, or rather allow them to coordinate on extra profit margins? Are the existing laws and regulations capable of protecting competition again any new threats that pricing algorithms may pose to it? How to detect when algorithms are harming welfare and are outside the boundary of law? Finding answers to these questions and a balanced attitude towards algorithmic pricing in the business practice are of the utmost social relevance and deserve the careful attention of competition policy professionals. At the core of concerns about such algorithms lies the question whether autonomous algorithmic collusion is possible at all. Can algorithms really learn to price supracompetitively, without engaging in illicit communication that would already be punishable as cartel law violation, and at the same time learn to price competitively against firms with different algorithms? Maybe algorithmic collusion is nothing more than science fiction. If on the other hand algorithmic collusion is implementable in reality, then competition authorities are right to jump on the subject and should address challenging questions such as how to detect and mitigate the threat.

Although we find that Q-learning has not been shown to collude, we certainly do believe that well-performing pricing algorithms can be constructed that do learn to collude. Yet claims of existence of practically relevant colluding pricing algorithms need to satisfy a few more criteria. Let us name the following five. First, because demand in practice is random, this should be modelled as a non-degenerate random variable in the way that is

common in the dynamic pricing literature since at least the work of Mills (1959). Second, the demand function – the expected demand as function of the selling prices – should be assumed unknown to the firms, for example in a Bayesian or frequentist, parametric or non-parametric manner. Algorithms cannot use information that is unknown to the user of that algorithm to construct, for example, a set of feasible prices. Third, to assess firms' choices, the profit-gain from collusion should be measured according to their actual objectives – typically expected discounted profit over an infinite time horizon or undiscounted expected total profit over a finite time horizon. Fourth, the firms should not be assumed to start their algorithms under exactly timed and synchronized conditions, since this inevitably would require some coordination. Fifth, to qualify as collusive, algorithms should not only generate supracompetitive prices and profits when playing against the same (or a similar) algorithm, but should also perform well against a class of reasonable 'competitive' alternative pricing algorithms.

This last point is crucial. The challenge is *not* to construct an algorithm that generates supracompetitive outcomes when playing against its own type. In a full information framework this is easy: always price at the monopolist' price – or at any other price with supracompetitive profits. In an incomplete information setting, one should augment this with demand learning, but it remains true: an algorithm that prices supra-competitively when playing against itself is not difficult to construct and its existence is not surprising.[27] What is needed is that the algorithm responds well if its supracompetitive prices are undercut by the competitor. Now, recent papers take a small step in this direction: *if* the competitor uses the *exact same* simple Q-learning algorithm, then a properly functioning reward-punishment scheme could deter the competitor from playing the Nash price – that is, if, for the sake of the argument, the limiting strategies indeed contain reward-punishment schemes. But if a competitor uses just a slightly different algorithm, there are no guarantees that Q-learning will respond well. In fact, it might respond very poorly, for example against Exp3.

We fail to see why a rational agent would use an algorithm, knowing that it responds very poorly to a simple and intuitive alternative algorithm that the competitor could use and benefit from. It might, if the firm already *knows* that the competitor is using the same algorithm and will continue to do so. But the challenge is showing that algorithmic collusion can be realized by a rational agent *without coordination with competitors*. That means that algorithmic collusion should be obtained *without knowing which algorithm the competitor will use*. Thus, in the absence of coordination, an algorithm that learns to price supra-competitively when playing against itself would *only be used by a rational agent* if the algorithm also guaranteed a good performance against *other reasonable algorithms*

[27] In fact, the work by Kirman (1975), recently revisited by Cooper, Homem-de Mello and Kleywegt (2015), already points in this direction.

that the competitor might use. Without offering such guarantees one misses the essence of the challenge to establish coordination-free algorithmic collusion.

This reasoning suggests that an algorithm that can learn to collude *and* simultaneously has good performance against reasonable alternatives, should include three modules: a collusive one, a competitive one, and a switch-mechanism to determine which module to use. To see this, consider the repeated prisoner's dilemma with known payoff matrix. If one player uses the simplistic algorithm of always playing C then the other player should always play D; if one always plays the best response then the other should always play D; and if one has a sophisticated collusive algorithm then both players may discover that they both benefit from playing C *and* know that they both do not use simplistic technology. To learn to collude, they both thus need to discover whether the supracompetitive action C by their opponent is generated by a simplistic algorithm or a sophisticated collusive algorithm; in the first case they should respond with D, in the second case with C. An algorithm that can learn to collude and simultaneously has good performance against reasonable alternatives likely needs to have these three components. The collusive module aims to learn the collusive price, the competitive module aims to respond optimally against the opponent, and the switch-mechanism aims to discover whether pricing supracompetitively outperforms pricing competitively against this opponent.

From this perspective, it also becomes clear why existing well-performing algorithms are *not* likely to collude and at the same time respond well against competitive players. Most existing well-performing algorithms learn to respond optimally to the environment that they are in, and will not easily converge to an action that can be improved upon. Thus, if its competitor(s) price(s) supra-competitively, the algorithm tries to learn a best response. Some sophistication is needed to learn to collude in the sense described above. It would, for example, require figuring out whether a competitor's supra-competitive price should be 'punished' by a best response, or whether it should be interpreted as an invitation to tacit collusion. To learn this, the algorithm should discover how the competitor responds to its actions, and it should take into account that its own actions may also be interpreted by the other to discover whether tacit collusion is possible. In addition, it should do all of this in a setting where the price-demand relation is unknown upfront and where the firms cannot coordinate on synchronization. That such algorithms can be constructed has recently been demonstrated by Meylahn and den Boer (2020) and Loots and den Boer (2022). But simple existing algorithms usually do not satisfy these requirements. Competition authorities that want to identify pricing algorithms with potential to be used in practice and collude best develop expertise to find these kinds of structures in their codes.

Finally, a note on the idea of intentionality of collusion. In some contributions to the

debate on algorithmic collusion, a distinction is made between 'autonomous' or 'spontaneous' algorithmic collusion versus 'pre-programmed' or 'intended' collusion. Based on our analyses, we believe this distinction requires some nuancing. The intentions of software developers may be relevant from the perspective of the legality of using their software, but from a programming perspective one cannot really talk about autonomous collusion by an algorithm. By careful studying the Bellman equations, one could have known in advance that collusive strategy-equilibria are implicitly present in Q-learning algorithms. This may have been programmed unintentionally by a perfectly benign programmer, but a malign software developer could have come up with exactly this algorithm to generate collusive outcomes - although he or she would have done a pretty bad job, as we show in this paper. Algorithms do not learn or act autonomously: they are simply doing what they are programmed to do – hence all (if any) algorithmic collusion is pre-programmed. The outcomes may be unexpected, unwanted, or illegal even, but they are not 'unintended' by the software. In other words, 'autonomous' or 'spontaneous' is not a property of an algorithm, but of the human perception of its workings. These qualifications of particular algorithms that learn to collude are confusing and should better be avoided.

References

Abada, Ibrahim, and Xavier Lambin. 2020. "Artificial Intelligence: Can Seemingly Collusive Outcomes Be Avoided?" *Management Science*, to appear.

Abada, Ibrahim, Xavier Lambin, and Nikolay Tchakarov. 2022. "Collusion by Mistake: Does Algorithmic Sophistication Drive Supra-Competitive Profits?" *Available at SSRN 4099361.*

Aoyagi, Masaki. 1998. "Mutual Observability and the Convergence of Actions in a Multi-Person Two-Armed Bandit Model." *Journal of Economic Theory*, 82: 405–424.

Asker, John, Chaim Fershtman, and Ariel Pakes. 2021. "Artificial intelligence and pricing: The impact of algorithm design." National Bureau of Economic Research Working Paper 28535.

Asker, John, Chaim Fershtman, and Ariel Pakes. 2022*a*. "Artificial Intelligence, Algorithm Design and Pricing." *AER, Papers and Proceedings*, 112: 452–456.

Asker, John, Chaim Fershtman, and Ariel Pakes. 2022*b*. "The Impact of AI Design on Pricing."

Assad, Stephanie, Emilio Calvano, Giacomo Calzolari, Robert Clark, Vincenzo Denicolò, Daniel Ershov, Justin Johnson, Sergio Pastorello, Andrew Rhodes,

Lei Xu, et al. 2021. "Autonomous algorithmic collusion: Economic research and policy implications." *Oxford Review of Economic Policy*, 37(3): 459–478.

Barfuss, Wolfram, and Janusz M. Meylahn. 2022. "Intrinsic fluctuations of reinforcement learning promote cooperation." *Available at SSRN 4207322*.

Beneke, Francisco, and Mark-Oliver Mackenrodt. 2020. "Remedies for algorithmic tacit collusion." *Journal of Antitrust Enforcement*, 9(1): 152–176.

Bernhardt, Lea, and Ralf Dewenter. 2020. "Collusion by code or algorithmic collusion? When pricing algorithms take over." *European Competition Journal*, 16(2-3): 312–342.

Bertini, Marco, and Oded Koenigsberg. 2021. "The Pitfalls of Pricing Algorithms: Be Mindful of How They Can Hurt Your Brand." *Harvard Business Review*, 99(5): 74–83.

Brown, Zach Y., and Alexander MacKay. 2021. "Competition in pricing algorithms." National Bureau of Economic Research.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicoló, and Sergio Pastorello. 2020*a*. "Artificial Intelligence, Algorithmic Pricing, and Collusion." *American Economic Review*, 110(10): 3267–97.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicoló, Joseph E Harrington Jr, and Sergio Pastorello. 2020*b*. "Protecting consumers from collusive prices due to AI." *Science*, 370(6520): 1040–1042.

Cartea, Álvaro, Patrick Chang, José Penalva, and Harrison Waldon. 2022. "The Algorithmic Learning Equations: Evolving Strategies in Dynamic Games." *Available at SSRN 4175239*.

CMA. 2021. "Algorithms: How they can reduce competition and harm consumers."

Coglianese, Cary, and Alicia Lai. 2022. "Antitrust by Algorithm." *Stanford Computational Antitrust*, 2: 1–23.

Cooper, William L., Tito Homem-de Mello, and Anton J. Kleywegt. 2015. "Learning and pricing with models that do not explicitly incorporate competition." *Operations Research*, 63(1): 86–103.

Epivent, Andréa, and Xavier Lambin. 2022. "On Algorithmic Collusion and Reward-Punishment Schemes." *Available at SSRN 4227229*.

Eschenbaum, Nicolas, Filip Mellgren, and Philipp Zahn. 2022. "Robust Algorithmic Collusion." *arXiv:2201.00345 [econ.GN]*.

Ezrachi, Ariel, and Maurice E. Stucke. 2016. *Virtual competition: the promise and perils of the algorithm-driven economy.* Cambridge, Massachusetts : Harvard University Press.

Ezrachi, Ariel, and Maurice E. Stucke. 2020. "Sustainable and Unchallenged Algorithmic Tacit Collusion." *Northwestern Journal of Technology and Intellectual Property*, 17(2): 217–260.

Gal, Michal S. 2019. "Algorithms as Illegal Agreements." *Berkeley Technology Law Journal*, 34(67).

Gal, Michal S. 2022. "Limiting Algorithmic Cartels." *Available at SSRN 4063081.*

Han, Bingyan. 2022. "Cooperation between independent market makers." *Quantitative Finance*, 22(11): 2005–2019.

Harrington, Jr, Joseph E. 2018. "Developing Competition Law for Collusion by Autonomous Price-Setting Agents." *Journal of Competition Law and Economics*, 14(3): 331–363.

Hettich, Matthias. 2021. "Algorithmic Collusion: Insights from Deep Learning." *Available at SSRN 3785966.*

Kastius, Alexander, and Rainer Schlosser. 2021. "Dynamic pricing under competition using reinforcement learning." *Journal of Revenue and Pricing Management*, 21(1): 50–63.

Kirman, Alan P. 1975. "Learning by firms about demand conditions." *In R. H. Day and T. Graves, editors, Adaptive Economic Models, pages 137–156. Academic Press, New York.*

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

Kühn, Kai-Uwe, and Steve Tadelis. 2018. "The Economics of Algorithmic Pricing: Is collusion really inevitable?" working paper.

Lattimore, Tor, and Csaba Szepesvári. 2020. *Bandit algorithms.* Cambridge University Press, Cambridge, United Kingdom; New York, NY.

Li, Hongmin, and Woonghee T. Huh. 2011. "Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications." *Manufacturing & Service Operations Management*, 13(4): 549–563.

Loots, Thomas, and Arnoud V. den Boer. 2022. "Data-driven collusion and competition in a pricing duopoly with multinomial logit demand." *Production and Operations Management*, forthcoming.

Mazundar, Aneesa. 2022. "Algorithmic Collusion: Reviving Section 5 of the FTC Act." *Columbia Law Review*, 122: 449–488.

Mehra, Salil K. 2016. "Antitrust and the Robo-Seller: Competition in the Time of Algorithms." *Minnesota Law Review*, 100: 1323–1375.

Mehra, Salil K. 2021. "Price Discrimination-Driven Algorithmic Collusion: Platforms for Durable Cartels." *Stanford Journal of Law, Business & Finance*, 26: 171–221.

Meylahn, Janusz M., and Arnoud V. den Boer. 2020. "Learning to Collude in a Pricing Duopoly." *Manufacturing & Service Operations Management*, forthcoming.

Meylahn, Janusz M., and Lars Janssen. 2022. "Limiting dynamics for Q-learning with memory one in symmetric two-player, two-action games." *Complexity*, 2022.

Mills, Edwin S. 1959. "Uncertainty and Price Theory." *The Quarterly Journal of Economics*, 73(1): 116–130.

OECD. 2017. "Algorithms and Collusion: Competition Policy in the Digital Age."

Rothschild, M. 1974. "A two-armed bandit theory of market pricing." *Journal of Economic Theory*, 9: 185–202.

Sánchez-Cartas, J. Manuel, and Evangelos Katsamakas. 2022. "Effects of Algorithmic Pricing on Platform Competition." *Available at SSRN 4027365.*

Schwalbe, Ulrich. 2019. "Algorithms, Machine Learning, and Collusion." *Journal of Competition Law & Economics*, 14(4): 568–607.

Van Seijen, Harm, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. 2009. "A theoretical and empirical analysis of Expected Sarsa." 177–184, IEEE.

Veljanovski, Cento. 2022. "Algorithmic Antitrust: A Critical Overview." In *Economic Analysis of Law in European Legal Scholarship.* , ed. Aurelien Portuese, 39–64. Cham:Springer.

Waltman, Ludo, and Uzay Kaymak. 2015. "Q-Learning Agents in a Cournot Oligopoly Model." *Journal of Economic Dynamics & Control*, 32: 3275–3293.

Wang, Qiaochu, Yan Huang, and Param Vir Singh. 2022. "Algorithms, Artificial Intelligence and Simple Rule Based Pricing." *Available at SSRN 4144905.*

Wang, Yanwen. 2022. "Will Algorithms Learn to Collude? Insights from the Natural Policy Gradient Method." *Department of Economics, University of Virginia.*

Wieting, Marcel, and Geza Sapi. 2021. "Algorithms in the Marketplace: An Empirical Analysis of Automated Pricing in E-Commerce." *Available at SSRN 3945137.*

Xiong, Wei, and Rama Cont. 2021. "Interactions of market making algorithms." *ICAIF '21.* New York, NY, USA:Association for Computing Machinery.

Appendix

A Additional numerical results: Effect of learning rate on fraction of WSLS and GT

In this section, we show how the fraction of limiting collusive strategies WSLS and GT depend on α. Table 5 summarizes how the fraction of strategies converging to WSLS and the fraction of strategies converging to GT depend on α.

α	0.05	0.1	0.15	0.2	0.25
WSLS	0.28 ± 0.03	0.50 ± 0.03	0.43 ± 0.03	0.39 ± 0.03	0.34 ± 0.03
GT	0.002 ± 0.002	0.03 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.02 ± 0.01

Table 5: The first and second row give the Wilson score interval for trajectories converging to the WSLS and GT strategy pairs respectively of 1000 trajectories for different values of α when using the baseline parameterization. Q-values are initialized as in Calvano et al. (2020a) and all trajectories converged. All trajectories converged.

B Additional numerical results: alternative feasible price sets

In this section, we consider two alternative feasible price sets which are symmetric around the Nash price, defined as

$$\tilde{\mathcal{A}}_i = \left\{ p_i^N - (\xi+1)\zeta_i + \frac{k}{m-1} \cdot 2(1+\xi)\zeta_i : k = 0, \ldots, m-1 \right\}$$
$$\hat{\mathcal{A}}_i = \left\{ c_i + \frac{k}{m+1} \cdot 2(p_i^N - c_i) : k = 1, \ldots, m \right\},$$

where $\zeta_i = p_i^M - p_i^N$.

In Table 6 we show how using $\tilde{\mathcal{A}}_i$ instead of $\mathcal{A}_i$ leads to negative $\tilde{\Delta}$ values and negative average profit when the competing player prices uniformly at random. This shows that any surplus derived within the effective time horizon (during which Q-learning is statistically indistinguishable from pricing uniformly-at-random) is a property of the price set.

In Table 7 we repeat the numerical experiments reported in Table 4 but now with feasible price set $\tilde{\mathcal{A}}_i$ instead of $\mathcal{A}_i$. The results show that Q-learning is still vulnerable when playing against Exp3, in particular if δ is close to one, but the surplus as measured by $\tilde{\Delta}$ is negative. This illustrates that the positive values of $\tilde{\Delta}$ in Table 4 should not be

interpreted as that Exp3 is a collusive strategy, but that they are simply a consequence of how the feasible price set is selected.

Price set	Average price	Average profit	Expected $\tilde{\Delta}$
$\mathcal{A}_i$	1.69895 (+15%)	0.279906 (+27%)	0.497357
$\tilde{\mathcal{A}}_i$	1.47293 (+0%)	0.164479 (-26%)	-0.510177
$\hat{\mathcal{A}}_i$	1.47293 (+0%)	0.172288 (-22%)	-0.442015

Table 6: Average price and profit when both firms set their price uniformly at random, for three feasible price sets. The first, $\mathcal{A}_i$, is the original price set when using $\xi = 0.1$, the second, $\tilde{\mathcal{A}}_i$, is symmetric around the Nash price with $\xi = 0$ and the last, $\hat{\mathcal{A}}_i$ starts just above the cost price and is centred around the Nash price. The numbers in between parentheses report the relative difference compared to the Nash price 1.47 and profit 0.22

	$\delta = 0.95$		$\delta = 0.99$	
	Exp3	Q-learning	Exp3	Q-learning
Exp3	$-0.48 \pm .005$	$-0.49 \pm .008$	$-0.43 \pm .005$	$-0.46 \pm .005$
Q-learning	$-0.50 \pm .007$	$-0.51 \pm .009$	$-0.48 \pm .005$	$-0.51 \pm .004$
	$\delta = 0.999$		$\delta = 0.9999$	
	Exp3	Q-learning	Exp3	Q-learning
Exp3	$-0.26 \pm .007$	$-0.37 \pm .003$	$-0.05 \pm .006$	$-0.22 \pm .002$
Q-learning	$-0.42 \pm .005$	$-0.51 \pm .001$	$-0.35 \pm .004$	$-0.49 \pm .0004$

Table 7: Empirically observed $\tilde{\Delta}$ in the baseline parametrization using $\tilde{\mathcal{A}}_i$ with $\xi = 0$ for all combinations of Exp3 and Q-learning and different values of δ. For Q-learning, we take $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$. For Exp3, we fix η at the optimal value given in Theorem 2. The values are an average over 1000 samples. In all cases, we give a 95% confidence interval. Each entry gives the value of payoff to the row player, normalized according to $\tilde{\Delta}$ to facilitate comparison between different discount factors. Values are computed based on the effective time horizon, which in each case is given by $T_{0.95} = 207$, $T_{0.99} = 1054$, $T_{0.999} = 10587$, $T_{0.9999} = 105921$.

C Proof of Theorem 1.

In this proof, we treat $\epsilon \in [0, 1]$ as a variable. For $i = \pm 1$, $(p_i, p_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$ and $\epsilon \in [0, 1]$ define

$$\pi_i(p_i, p_{-i}; \epsilon) := (1 - \epsilon)\pi_i(p_i, p_{-i}) + \frac{\epsilon}{|\mathcal{A}_{-i}|} \sum_{p \in \mathcal{A}_{-i}} \pi_i(p_i, p).$$

For $i = \pm 1$, $(\sigma^{(i)}, \sigma^{(-i)}) \in \Sigma_i \times \Sigma_{-i}$ and $\epsilon \in [0, 1]$, define

$$V^{(i)}_{\sigma^{(i)}, \sigma^{(-i)}}(s_i, s_{-i}; \epsilon) := \sum_{t=1}^{\infty} \delta^{t-1} \mathbb{E}[\pi_i(p_i(t), p_{-i}(t); \epsilon) | s(1) = (s_i, s_{-i})],$$

	$\delta = 0.95$		$\delta = 0.99$	
	Exp3	Q-learning	Exp3	Q-learning
Exp3	$-0.42 \pm .004$	$-0.43 \pm .007$	$-0.38 \pm .005$	$-0.41 \pm .004$
Q-learning	$-0.44 \pm .006$	$-0.45 \pm .009$	$-0.42 \pm .004$	$-0.44 \pm .004$
	$\delta = 0.999$		$\delta = 0.9999$	
	Exp3	Q-learning	Exp3	Q-learning
Exp3	$-0.24 \pm .007$	$-0.33 \pm .003$	$-0.06 \pm .006$	$-0.21 \pm .002$
Q-learning	$-0.36 \pm .005$	$-0.44 \pm .001$	$-0.30 \pm .004$	$-0.42 \pm .0004$

Table 8: Empirically observed $\tilde{\Delta}$ in the baseline parametrization using $\hat{\mathcal{A}}_i$ for all combinations of Exp3 and Q-learning and different values of δ. For Q-learning, we take $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$. For Exp3, we fix η at the optimal value given in Theorem 2. The values are an average over 1000 samples. In all cases, we give a 95% confidence interval. Each entry gives the value of payoff to the row player, normalized according to $\tilde{\Delta}$ to facilitate comparison between different discount factors. Values are computed based on the effective time horizon, which in each case is given by $T_{0.95} = 186$, $T_{0.99} = 947$, $T_{0.999} = 9514$, $T_{0.9999} = 95181$.

where $(p_i(t), p_{-i}(t)) = (\sigma^{(i)}(s_i(t), s_{-i}(t)), \sigma^{(-i)}(s_{-i}(t), s_i(t))$ and $(s_i(t+1), s_{-i}(t+1)) = (p_i(t), p_{-i}(t))$ for all $t \in \mathbb{N}$. By the 'principle of optimality',

$$V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(s_i, s_{-i}; \epsilon) = \pi_i(\sigma^{(i)}(s_i, s_{-i}), \sigma^{(-i)}(s_{-i}, s_i); \epsilon) + \delta V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(\sigma^{(i)}(s_i, s_{-i}), \sigma^{(-i)}(s_{-i}, s_i); \epsilon).$$

Let

$$A^{\sigma^{(i)},\sigma^{(-i)}}_{(s_i,s_{-i}),(p_i,p_{-i})} := \mathbf{1}\{\sigma^{(i)}(s_i, s_{-i}) = p_i \text{ and } \sigma^{(-i)}(s_{-i}, s_i) = p_{-i}\},$$

and let $A^{\sigma^{(i)},\sigma^{(-i)}}$ be the $m^2 \times m^2$ matrix with rows and columns both arranged lexicographically in the order of prices from low to high; i.e., if $p_i^{(1)} < \ldots < p_i^{(m)}$ are the feasible prices of player i ordered from low to high, then $A^{\sigma^{(i)},\sigma^{(-i)}}_{(p_i^{(k_i)},p_{-i}^{(k_{-i})}),(p_i^{\ell_i}),p_{-i}^{(\ell_{-i})})}$ is the element at row $(k_i - 1)m + k_{-i}$ and column $(\ell_i - 1)m + \ell_{-i}$ of $A^{\sigma^{(i)},\sigma^{(-i)}}$. Because $A^{\sigma^{(i)},\sigma^{(-i)}}$ is a right-stochastic matrix, all eigenvalues have absolute value at most one. From $\delta \in (0, 1)$ it follows that $I - \delta A^{\sigma^{(i)},\sigma^{(-i)}}$ is invertible, and

$$\begin{pmatrix} V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(p_i^{(1)}, p_{-i}^{(1)}; \epsilon) \\ V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(p_i^{(1)}, p_{-i}^{(2)}; \epsilon) \\ \vdots \\ V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(p_i^{(m)}, p_{-i}^{(m)}; \epsilon) \end{pmatrix} = (I - \delta A_{\sigma^{(i)},\sigma^{(-i)}})^{-1} \begin{pmatrix} \pi_i(p_i^{(1)}, p_{-i}^{(1)}; \epsilon) \\ \pi_i(p_i^{(1)}, p_{-i}^{(2)}; \epsilon) \\ \vdots \\ \pi_i(p_i^{(m)}, p_{-i}^{(m)}; \epsilon) \end{pmatrix}.$$

It follows that, for all $\epsilon, \epsilon' \in [0,1]$ and all $(s_i, s_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$,

$$\begin{aligned}
&\left|V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(s_i, s_{-i}; \epsilon) - V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(s_i, s_{-i}; \epsilon')\right| \\
\leq &\left|\left|V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(\cdot, \cdot; \epsilon) - V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(\cdot, \cdot; \epsilon')\right|\right| \\
\leq &\left|\left|(I - \delta A_{\sigma^{(i)},\sigma^{(-i)}})^{-1}\right|\right| \cdot ||\pi_i(\cdot, \cdot; \epsilon) - \pi_i(\cdot, \cdot; \epsilon')|| \\
\leq &\frac{1}{1-\delta} \cdot m \cdot \sup_{(p_i, p_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}} \left|(\epsilon' - \epsilon)\pi_i(p_i, p_{-i}) + \frac{\epsilon - \epsilon'}{|\mathcal{A}_{-i}|} \sum_{p \in \mathcal{A}_{-i}} \pi_i(p_i, p)\right| \\
\leq &\frac{\pi_{\max} \cdot m}{1-\delta}|\epsilon - \epsilon'|.
\end{aligned}$$

Let $V^{(i)}_{\sigma^{(-i)}}(s_i, s_{-i}; \epsilon) := \max_{\sigma^{(i)} \in \Sigma_i} V^{(i)}_{\sigma^{(i)},\sigma^{(-i)}}(s_i, s_{-i}; \epsilon)$. We have

$$V^{(i)}_{\sigma^{(-i)}}(s_i, s_{-i}; 1) = (1-\delta)^{-1} \frac{1}{|\mathcal{A}_{-i}|} \sum_{p_{-i} \in \mathcal{A}_{-i}} \pi(p^*, p_{-i}),$$

for all states $(s_i, s_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$ and all opponent's strategies $\sigma^{(-i)} \in \Sigma_{-i}$. Therefore, for all $p \in \mathcal{A}_i$, $p \neq p^*$,

$$\begin{aligned}
&\frac{1}{|\mathcal{A}_{-i}|} \sum_{p_{-i} \in \mathcal{A}_{-i}} \{\pi_i(p^*, p_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(p^*, p_{-i}; 1)\} \\
&- \frac{1}{|\mathcal{A}_{-i}|} \sum_{p_{-i} \in \mathcal{A}_{-i}} \{\pi_i(p, p_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(p, p_{-i}; 1)\} \\
&= \frac{1}{|\mathcal{A}_{-i}|} \sum_{p_{-i} \in \mathcal{A}_{-i}} \{\pi_i(p^*, p_{-i}) - \pi_i(p, p_{-i})\} \geq \kappa,
\end{aligned}$$

for all opponent's strategies $\sigma^{(-i)} \in \Sigma_{-i}$, where $\kappa := \min_{p \in \mathcal{A}_i, p \neq p^*} \frac{1}{|\mathcal{A}_{-i}|} \sum_{p_{-i} \in \mathcal{A}_{-i}} \{\pi_i(p^*, p_{-i}) - \pi_i(p, p_{-i})\}$. In addition, for any state $(s_i, s_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$, we have

$$(1-\delta)^{-1}\pi^{(i)}_{\min}(\epsilon) \leq V^{(i)}_{\sigma^{(-i)}}(s_i, s_{-i}; \epsilon) \leq (1-\delta)^{-1}\pi^{(i)}_{\max}(\epsilon),$$

where $\pi^{(i)}_{\min}(\epsilon) := \min_{p_i \in \mathcal{A}_i, p_{-i} \in \mathcal{A}_{-i}} \pi_i(p_i, p_{-i}; \epsilon)$ and $\pi^{(i)}_{\max}(\epsilon) := \max_{p_i \in \mathcal{A}_i, p_{-i} \in \mathcal{A}_{-i}} \pi_i(p_i, p_{-i}; \epsilon)$. Combining the above, we obtain that, for all $p \in \mathcal{A}_i$, $p \neq p^*$, and all states $(s_i, s_{-i}) \in$

$\mathcal{A}_i \times \mathcal{A}_{-i}$

$$
\begin{aligned}
&\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{\pi_i(p^*,p_{-i})+\delta V^{(i)}_{\sigma^{(-i)}}(p^*,p_{-i};\epsilon)\}\\
+&(1-\epsilon)\{\pi_i(p^*,\sigma^{(-i)}(s_{-i},s_i))+\delta V^{(i)}_{\sigma^{(-i)}}(p^*,\sigma^{(-i)}(s_{-i},s_i);\epsilon)\}\\
-&\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{\pi_i(p,p_{-i})+\delta V^{(i)}_{\sigma^{(-i)}}(p,p_{-i};\epsilon)\}\\
-&(1-\epsilon)\{\pi_i(p,\sigma^{(-i)}(s_{-i},s_i))+\delta V^{(i)}_{\sigma^{(-i)}}(p,\sigma^{(-i)}(s_{-i},s_i);\epsilon)\}\\
\geq&\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{\pi_i(p^*,p_{-i})+\delta V^{(i)}_{\sigma^{(-i)}}(p^*,p_{-i};1)\}\\
-&\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{\pi_i(p,p_{-i})+\delta V^{(i)}_{\sigma^{(-i)}}(p,p_{-i};1)\}\\
+&(1-\epsilon)\{\pi_i(p^*,\sigma^{(-i)}(s_{-i},s_i))-\pi_i(p,\sigma^{(-i)}(s_{-i},s_i))\}\\
+&(1-\epsilon)\delta\{V^{(i)}_{\sigma^{(-i)}}(p^*,\sigma^{(-i)}(s_{-i},s_i);\epsilon)-V^{(i)}_{\sigma^{(-i)}}(p,\sigma^{(-i)}(s_{-i},s_i);\epsilon)\}\\
-&\epsilon\delta\frac{1}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{V^{(i)}_{\sigma^{(-i)}}(p^*,p_{-i};1)-V^{(i)}_{\sigma^{(-i)}}(p^*,p_{-i};\epsilon)\}\\
-&\epsilon\delta\frac{1}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{V^{(i)}_{\sigma^{(-i)}}(p,p_{-i};\epsilon)-V^{(i)}_{\sigma^{(-i)}}(p,p_{-i};1)\}\\
\geq&\epsilon\kappa+(1-\epsilon)\min_{p\in\mathcal{A}_i,\tilde{p}\in\mathcal{A}_{-i}}\{\pi_i(p^*,\tilde{p})-\pi_i(p,\tilde{p})\}\\
-&(1-\epsilon)\delta(1-\delta)^{-1}(\pi^{(i)}_{\max}(\epsilon)-\pi^{(i)}_{\min}(\epsilon))-2\epsilon\delta\frac{\pi_{\max}\cdot m}{1-\delta}(1-\epsilon),
\end{aligned}
$$

which is strictly greater than zero if ϵ is sufficiently close to one. As a result, p^* is the optimal action in all states. This completes the proof of the theorem.

D Proof of Theorem 2.

To prove the theorem, we adapt Section 11.4 of Lattimore and Szepesvári (2020) to the setting with infinite-horizon discounted rewards. For all $t \in \mathbb{N}\cup\{0\}$ and $p \in \mathcal{A}_i$ let

$$
\hat{S}^{(i)}_{t,p} := \sum_{s=1}^{t}\delta^s \hat{X}^{(i)}_{s,p}, \qquad \hat{S}^{(i)}_{t} := \sum_{s=1}^{t}\delta^s \sum_{p\in\mathcal{A}_i} P^{(i)}_{s,p}\hat{X}^{(i)}_{s,p},
$$

and

$$
W_t := \sum_{p\in\mathcal{A}_i}\exp(\eta\hat{S}^{(i)}_{t,p}),
$$

where an empty sum is defined as zero and hence $\hat{S}_{0,p}^{(i)} = 0$ and $W_0 = |\mathcal{A}_i|$. For all $t \in \mathbb{N}$,

$$\begin{aligned}\frac{W_t}{W_{t-1}} &= \sum_{p\in\mathcal{A}_i} \frac{\exp(\eta \hat{S}_{t-1,p}^{(i)})}{W_{t-1}} \exp(\eta\delta^t \hat{X}_{t,p}^{(i)}) = \sum_{p\in\mathcal{A}_i} P_{t,p}^{(i)} \exp(\eta\delta^t \hat{X}_{t,p}^{(i)}) \\ &\leq \sum_{p\in\mathcal{A}_i} P_{t,p}^{(i)} (1 + \eta\delta^t \hat{X}_{t,p}^{(i)} + \eta^2\delta^{2t} (\hat{X}_{t,p}^{(i)})^2) \\ &\leq \exp\left(\eta\delta^t \sum_{p\in\mathcal{A}_i} P_{t,p}^{(i)} \hat{X}_{t,p}^{(i)} + \eta^2\delta^{2t} \sum_{p\in\mathcal{A}_i} P_{t,p}^{(i)} (\hat{X}_{t,p}^{(i)})^2\right),\end{aligned}$$

where the first inequality uses $\exp(x) \leq 1 + x + x^2$ for all $x \leq 1$ – note that $\hat{X}_{t,p}^{(i)} \leq 1$ –, and the second inequality uses $1 + x \leq \exp(x)$ for all $x \in \mathbb{R}$. It follows that

$$\begin{aligned}\exp(\eta \hat{S}_{t,p}^{(i)}) \leq W_t &= W_0 \prod_{s=1}^{t} \frac{W_s}{W_{s-1}} \\ &\leq |\mathcal{A}_i| \cdot \exp\left(\sum_{s=1}^{t} \eta\delta^s \sum_{p\in\mathcal{A}_i} P_{s,p}^{(i)} \hat{X}_{s,p}^{(i)} + \sum_{s=1}^{t} \eta^2\delta^{2s} \sum_{p\in\mathcal{A}_i} P_{s,p}^{(i)} (\hat{X}_{s,p}^{(i)})^2\right),\end{aligned}$$

and therefore, by taking logarithms and dividing by η,

$$\hat{S}_{t,p}^{(i)} - \hat{S}_t^{(i)} \leq \frac{\log(|\mathcal{A}_i|)}{\eta} + \eta \sum_{s=1}^{t} \delta^{2s} \sum_{p\in\mathcal{A}_i} P_{s,p}^{(i)} (\hat{X}_{s,p}^{(i)})^2. \tag{7}$$

Now

$$\begin{aligned}&\mathbb{E}\left[\sum_{s=1}^{t} \delta^{2s} \sum_{p\in\mathcal{A}_i} P_{s,p}^{(i)} (\hat{X}_{s,p}^{(i)})^2\right] = \sum_{s=1}^{t} \delta^{2s} \mathbb{E}\left[\sum_{p\in\mathcal{A}_i} P_{s,p}^{(i)} \left(1 - \frac{Y_s^{(i)} \cdot \mathbf{1}\{p_i(s) = p\}}{P_{s,p}^{(i)}}\right)^2\right] \\ &= \sum_{s=1}^{t} \delta^{2s} \mathbb{E}\left[1 - 2\sum_{p\in\mathcal{A}_i} Y_s^{(i)} \mathbf{1}\{p_i(s) = p\} + \sum_{p\in\mathcal{A}_i} \frac{(Y_s^{(i)})^2 \cdot \mathbf{1}\{p_i(s) = p\}}{P_{s,p}^{(i)}}\right] \\ &= \sum_{s=1}^{t} \delta^{2s} \mathbb{E}\left[1 - 2Y_s^{(i)} + \sum_{p\in\mathcal{A}_i} (Y_s^{(i)})^2\right] \\ &\leq \sum_{s=1}^{t} \delta^{2s} |\mathcal{A}_i| = \frac{\delta^2}{1-\delta^2} |\mathcal{A}_i|,\end{aligned} \tag{8}$$

where the inequality follows from $Y_s^{(i)} \in [0,1]$, and where the third equality uses $\sum_{p\in\mathcal{A}_i} Y_s^{(i)} \mathbf{1}\{p_i(s) = p\} = Y_s^{(i)}$ and, by conditioning on the action selected in period t,

$$\begin{aligned}
&\mathbb{E}\left[\frac{(Y_s^{(i)})^2 \cdot \mathbf{1}\{p_i(s)=p\}}{P_{s,p}^{(i)}} \mid \{p_i(s'), p_{-i}(s'), Y_{s'}^{(i)} : 1 \le s' < s\}\right] \\
=&\mathbb{E}\left[(Y_s^{(i)})^2 \mid \{p_i(s'), p_{-i}(s'), Y_{s'}^{(i)} : 1 \le s' < s\}\right],
\end{aligned}$$

We have

$$\mathbb{E}[\hat{X}_{s,p}^{(i)}] = 1 - \mathbb{E}[\hat{Y}_{t,p}^{(i)} | p_i(t) = p] P_{t,p}^{(i)} - \mathbb{E}[\hat{Y}_{t,p}^{(i)} | p_i(t) \neq p](1 - P_{t,p}^{(i)}) = \mathbb{E}[\pi_i(p, p_{-i}(s))],$$

for all $p \in \mathcal{A}_i$, and

$$\mathbb{E}[\pi_i(p_i(s), p_{-i}(s))] = \mathbb{E}[\sum_{p\in\mathcal{A}_i} P_{s,p}^{(i)} \hat{X}_{s,p}^{(i)}].$$

By (7) and (8),

$$\begin{aligned}
\text{Regret}_i &= \max_{p\in\mathcal{A}_i} \left\{ \sum_{s=1}^{t} \delta^s \mathbb{E}[(\pi_i(p, p_{-i}(s))] - \sum_{s=1}^{t} \delta^t \mathbb{E}[\pi_i(p_i(s), p_{-i}(s))] \right\} \\
&= \max_{p\in\mathcal{A}_i} \left\{ \mathbb{E}[\hat{S}_{t,p}^{(i)} - \hat{S}_t^{(i)}] \right\} \\
&\le \frac{\log(|\mathcal{A}_i|)}{\eta} + \eta |\mathcal{A}_i| \frac{\delta^2}{1-\delta^2} \\
&= 2\sqrt{\frac{\delta^2}{1-\delta^2} |\mathcal{A}_i| \log(|\mathcal{A}_i|)},
\end{aligned}$$

with $\eta = \sqrt{\frac{\log(|\mathcal{A}_i|)(1-\delta)^2}{\delta^2 |\mathcal{A}_i|}}$. This completes the proof.