ALGORITHMIC COLLUSION WITH IMPERFECT MONITORING[†]

Emilio Calvano[*‡§], Giacomo Calzolari[& §], Vincenzo Denicolò[*§] and Sergio Pastorello[*]

Janunary 2021

We show that if they are allowed enough time to complete the learning, Q-learning algorithms can learn to collude in an environment with imperfect monitoring adapted from Green and Porter (1984), without having been instructed to do so, and without communicating with one another. Collusion is sustained by punishments that take the form of "price wars" triggered by the observation of low prices. The punishments have a finite duration, being harsher initially and then gradually fading away. Such punishments are triggered both by deviations and by adverse demand shocks.

Keywords: Artificial Intelligence, Q-Learning, Imperfect Monitoring, Collusion.

J.E.L. codes: L41, L13, D43, D83.

1. INTRODUCTION

In the modern economy, firms increasingly delegate important strategic choices to algorithms of various sorts. Some, powered by Artificial Intelligence (AI), are capable of learning autonomously by adapting their behavior to past experience. This has raised concerns that such reinforcement-learning algorithms may learn to collude without having been specifically instructed to do so and without communicating with one another.[1]

To assess these concerns, two routes have proved useful. One tries to detect algorithmic collusion from market outcomes. For example, Assad et al. (2020) analyze German retail gasoline markets finding that the adoption of AI-pricing algorithms is associated with a sizeable increase in the stations' margins. The other approach simulates the algorithms' behavior under controlled conditions in artificial markets. For example, Klein (2019) conducts this kind of analysis for the Maskin and Tirole (1988) model of staggered pricing,

[1]This possibility would pose a challenge for antitrust policy, for reasons discussed in greater detail in Calvano, Calzolari, Denicolò, Harrington and Pastorello (2020).

and our previous work (Calvano et al., 2020) considers an infinitely repeated Bertrand game where firms can perfectly observe rivals' prices.[2] Both studies conclude that algorithmic collusion can arise spontaneously in such environments, but both recognize that more work is needed to ascertain the robustness of this finding.

In this paper, we contribute to this latter strand of research by analyzing the case of imperfect monitoring. Previous research has focused on the case of perfect monitoring because pricing algorithms are often used in marketplaces, such as Amazon, where each seller can monitor rivals' prices in real time, and because competition authorities stress that such markets are more vulnerable to collusion.[3] However, theory shows that collusion is possible even under imperfect monitoring, and algorithms are increasingly used also in markets where rivals' strategies are not easy to observe. One example is financial markets where agents exploit their inside information by hiding behind noise traders.[4] Another example is markets for electricity.[5] It is therefore important to study whether algorithmic collusion is possible when agents can observe aggregate outcomes (e.g., market prices) but not individual behaviors.[6]

While the examples mentioned above present important specificities, here we develop the analysis at an abstract level. We use Green and Porter's (1984) classic model of repeated interaction under imperfect monitoring, where firms set quantities and observe the price level but cannot perfectly infer rivals' outputs because demand is stochastic. This simple model is well understood and has become the workhorse of the theory of collusion with imperfect monitoring.[7]

To simplify the analysis, we assume that all firms use similar algorithms. In particular, we focus on algorithms of the Q-learning type, as we did in Calvano et al. (2020). Maintaining

[2]See also Johnson, Rodhes and Wildenbeest (2020).

[3]For example, the US Horizontal Merger Guidelines state:

> A market typically is more vulnerable to coordinated conduct if each competitively important firm's significant competitive initiatives can be promptly and confidently observed by that firm's rivals. This is more likely to be the case if the terms offered to customers are relatively transparent.

See https://www.justice.gov/sites/default/files/atr/legacy/2010/08/19/hmg-2010.pdf

[4]See for instance Foster and Viswanathan (1996) and Back, Cao and Willard (2000). For an analysis of collusion in the foreign exchange market, see Colla, Distaso and Vitale (2020).

[5]See Abada and Lambin (2020). This paper considers a model of "imperfect monitoring" with no uncertainty. As a result, even if firms cannot observe rivals' behavior directly, they can infer it from the observation of market outcomes.

[6]The case of imperfect monitoring is also interesting for policy reasons. One possible route to avoid algorithmic collusion is, indeed, to suppress some available information, for instance about rivals' prices or outputs. Our analysis sheds light on the effectiveness of this regulatory approach.

[7]See for instance Abreu, Pearce and Stacchetti (1986, 1990).

the focus on the same type of algorithms helps compare the results and identify the specific effects of imperfect monitoring.

We show that if Q-learning pricing algorithms are allowed enough time to complete the learning, they can collude even with imperfect monitoring. Collusion is not full, which is natural given that the algorithms learn purely by trial and error, but still yields substantial supra-competitive profits. As monitoring becomes more imperfect, making it more costly to punish deviations that can be confounded with adverse demand shocks, the level of profit decreases.

The strategies that the algorithms eventually learn are remarkably similar to those considered by Green and Porter (1984). That is, when the price level falls below a certain threshold, the algorithms enter into a "price war" that lasts for several periods and then revert to the pre-deviation output. One notable difference between the fully rational agents of Green and Porter (1984) and our algorithms, however, is that the former possess an infinitely long memory and thus know when to end the punishment, whereas the memory of the algorithms is short-lived by design. Plainly, the boundedness of memory hinders the direct implementation of punishments of duration longer than the memory. However, the algorithms learn to circumvent this problem by resorting to an ingenious method, which will be described in greater detail below.

We take these findings as an indication that imperfect monitoring is not, in itself, an insurmountable obstacle to autonomous algorithmic collusion. Naturally, our "experimental" approach can only show that algorithmic collusion is a real possibility but cannot provide conclusive evidence about its occurrence in practice. Skeptics may point to a number of specific modelling choices made in our simulations and wonder whether the results are robust to changes in these choices. Only more work can dispel, or substantiate, these doubts. This paper focuses on a single factor, namely, imperfect monitoring. Others remain to be addressed.

The practical significance of our findings may also be questioned on the ground that it takes a large number of periods, in the order of hundreds of thousands, for the algorithms to stabilize their behavior.[8] In fact, the algorithms start to collude much earlier than they converge to a limit strategy. Moreover, pricing algorithms are often trained off-line before being put to work, which may significantly reduce the time needed to learn. Yet, the time

[8]Q-learning algorithms learn slowly by design, as will be explained later. Other existing algorithms learn faster, and still faster ones may be developed in the future. Here we focus on Q-learning algorithms because they are well understood and characterized by few parameter whose economic interpretation is clear. More sophisticated algorithms, in contrast, require modeling choices that are quite arbitrary from an economic point of view and in this respect resembles "black boxes."

scale is another important issue to be addressed by future research. (It may be less of an issue in such markets as the foreign exchange market, where large volumes are traded at high frequencies. In these markets, a "period" may be a fraction of a second, so even millions of repetitions could take place in a few weeks.)

The rest of the paper is organized as follows. The next section provides a brief description of the Q-learning algorithms used in our simulations. Section 3 describes the economic environments where the simulations are performed. Section 4 discusses the results for the baseline case. Section 5 reports on various robustness exercises. Section 6 offers some concluding remarks.

2. Q-LEARNING

To facilitate comparison with previous work, we focus on Q-learning algorithms.[9] In this section, we shall first provide a short, abstract description of these algorithms[10] and then specify how they are implemented in our setting.

Algorithm i is trained to maximize:

$$E\left[\sum_{t=0}^{\infty}\delta^t \pi_{it}\right], \tag{1}$$

where π_{it} is the period-t profit of firm i and $\delta < 1$ is the discount factor. In each period, the algorithm observes a state variable $s_t \in S$ and then chooses an action $a_{it} \in A_i$. The reward π_{it} depends on the algorithm's own action a_{it}, on the actions of the other algorithms, $\mathbf{a}_{-i,t}$, and on the state s_t, possibly in a stochastic way. (In our specific setting, the state is payoff irrelevant but may nevertheless matter if players coordinate on a non-Markovian equilibrium). Given the current state s_t and the vector of actions $\mathbf{a}_t$, the game moves on to the next period where the new state is s_{t+1}

Q-learning is essentially a method for finding an optimal policy with no prior knowledge of the inherent structure of the game (i.e., the probability distribution that maps $(s_t, \mathbf{a}_t)$ into the reward π_{it} and the next state s_{t+1}). The method works by iteratively estimating the Q-function $Q_i(s, a_i)$, which represents the cumulative discounted payoff of taking action

[9]Other analyses of collusion among Q-learning algorithms include Waltman and Kaymak (2008), Klein (2019), Calvano et al. (2020), Johnson, Rodhes and Wildenbeest (2020) and Abada and Lambin (2020).

[10]We refer the reader to Sutton and Barto (2018) for a more in-depth treatment.

a_i in state s. This function may be defined recursively as follows:

$$Q_i(s, a_i) = E(\pi|s, a_i) + \delta E[\max_{a_i' \in A_i} Q(s', a_i')|s, a_i], \tag{2}$$

where a prime is a shorthand for the next-period value. In its simplest incarnation, Q-learning assumes that the sets S and A_i are finite and time invariant, and that the sets A_i are not state-dependent. For this case, the Q-function for player i becomes an $|S| \times |A_i|$ matrix.

To estimate this matrix, a Q-learning algorithm starts from an arbitrary initial matrix $\mathbf{Q}_{i0}$ and updates it on the basis of the information that accrues as the game is being played. In particular, after choosing action a_{it} in state s_t, the algorithm observes π_{it} and s_{t+1} and updates the corresponding cell of the matrix $Q_{it}(s, a_i)$ for $s = s_t$, $a_i = a_{it}$, according to the learning equation:

$$Q_{it+1}(s, a_i) = (1-\alpha)Q_{it}(s, a_i) + \alpha \left[\pi_{it} + \delta \max_{a_i' \in A_i} Q_{it}(s', a_i')\right]. \tag{3}$$

For all other cells $s \neq s_t$ and $a_i \neq a_{it}$, the Q-value does not change: $Q_{it+1}(s, a_i) = Q_{it}(s, a_i)$.[11] Equation (3) says that for the cell visited the new Q-value is a convex combination of the previous value and the current reward plus the discounted value of the state that is reached next. The weight $\alpha \in [0, 1]$ is the learning rate.

To approximate the true Q-matrix starting from an arbitrary matrix $\mathbf{Q}_{i0}$, the algorithm must experiment by selecting actions that may appear sub-optimal in the light of the knowledge acquired in the past. Various patterns of exploration may be considered. Here we focus on the so-called ε-greedy model of exploration, where the algorithm chooses the action with the highest Q-value in the relevant state, also known as the "greedy" action, with a probability $1 - \varepsilon$ and randomizes uniformly across all possible actions with probability ε. Thus, $1 - \varepsilon$ is the fraction of times the algorithm is in *exploitation mode*, while ε is the fraction of times it is in *exploration mode.*

Given their lack of prior knowledge of the problem at hand, initially the algorithms must explore widely. As time passes, however, the benefits from exploration decrease and the algorithms may spend more time in exploitation mode. Accordingly, we posit a time-

[11]The fact that only one cell of the matrix is updated in each period is the reason why Q-learning is slow. More sophisticated algorithms update also the Q-value of states close to the current one after each iteration.

declining exploration rate:

$$\varepsilon_t = e^{-\beta t}, \tag{4}$$

where $\beta > 0$ is a design parameter to be set by the researcher. This implies that initially the algorithms choose in purely random fashion,[12] but they then make the greedy choice more and more frequently. The greater β, the faster exploration vanishes.

3. ECONOMIC ENVIRONMENT

Following Green and Porter (1984), we model imperfect monitoring by considering a Cournot oligopoly with stochastic demand. In each period, firms choose their outputs after observing the past price. However, firms cannot observe rivals' past outputs, nor can they perfectly infer these outputs from the realized price, as demand is stochastic.

3.1. *Demand and costs*

Consider n firms that supply a homogeneous product with demand function

$$p_t = d_t - (q_{1t} + ... + q_{nt}), \tag{5}$$

where the stochastic demand parameter d_t is i.i.d. over time. The symmetric, constant marginal costs are normalized to zero. Thus, the per-period reward accruing to firm i is $\pi_{it} = p_t q_{it}$.

Initially, we shall focus on the case of duopoly, $n = 2$. We set the average value of d_t to 300. The corresponding individual output at this average demand is $q_i^C = 100$ in the non-cooperative Cournot equilibrium, and $q_i^M = 75$ with perfect collusion. The associated profits are $\pi^C = 10,000$ and $\pi^M = 11,250$, respectively.

Firms choose their output before demand realizes. This implies that uncertainty does not affect equilibrium outputs and profits, as long as the average demand level stays constant.

[12]In keeping with this assumption, we initialize the Q-matrix $\mathbf{Q}_{i0}$ at the discounted payoff that would accrue to algorithm i if competitors randomized uniformly, and we assume that the initial state s_0 is selected randomly.

3.2. *Strategies*

Generally speaking, a strategy for algorithm i is a function that maps the set of states S into the set of actions A_i. Different strategic environments may be obtained by making different assumptions about the state space.

Perfect monitoring is when each firm observes the rival's past output levels and may therefore condition its current choice on them. Assuming a one-period memory,[13] the corresponding state s_t is a pair $\{q_{1t-1}, q_{2t-1}\}$.

Imperfect monitoring, in contrast, is when firms observe only the past price level, which may not fully reveal the rival's output as demand is stochastic. Continuing to focus on the case of one-period memory, the state for player i then becomes $s_{it} = \{q_{it-1}, p_{t-1}\}$.[14] The literature on imperfect monitoring has however shown that collusive outcomes can also be supported when firms condition their current outputs on the past price only. Here we focus on simple strategies of this type, where the state is the previous period's price level, $s_t = p_{t-1}$.

3.3. *Discretization*

Since Q-learning requires a finite action and state space, we must discretize the above model. As for the firms' feasible actions, we assume that $A_i = \left\{q^1, q^2, ..., q^k\right\}$ with $q^{j+1} - q^j = v$. This means that q_{it} can take on k equally spaced values, with a step size of v.

Since under imperfect monitoring the state is the past price, we must also guarantee that the set of possible prices is finite. To this end, we assume that d_t can take on only h equally spaced values $\left\{d^1, d^2, ..., d^h\right\}$, with $d^{k+1} - d^k = mv$ for some integer m. The fact that the difference $d^{k+1} - d^k$ is a multiple of v implies that demand shocks may be confounded with changes in the rival's output, as both could result in the same price. This is the case, in particular, for intermediate values of the price. Very high or very low prices, in contrast, are fully revealing, i.e., observation of the price and knowledge of own output allows firms to infer the rival's output.[15]

[13] A finite memory is necessary for the state space to be finite and time invariant. Among the extensions, we shall consider the case of 2- or 3-period memory. Obviously, with a longer memory the set of states, and hence the Q matrix, become larger.

[14] In this case, the set of states is firm specific.

[15] The extent to which monitoring is imperfect can be measured by the fraction of possible price levels that do not fully reveal the rival's output, which under our assumptions is $\frac{(h-3)m+k(n-1)-n+2}{(h-1)m+k(n-1)-n+2}$. For $h = n = 2$ that simplifies to $\frac{k-m}{k+m}$.

3.4. *Baseline specification*

In the baseline specification of the model, we set $n = 2$ (a duopoly), $k = 15$ (the feasible output levels) and $h = 2$ (the possible realizations of demand). The two levels of demand are taken to be $d^1 = 290$ or $d^2 = 310$. These two values are assumed to be equally likely, and the demand shocks uncorrelated. The discount factor δ is 0.95.

We specify the sets A_i so that firms may choose output levels that are higher than the non-cooperative Cournot outputs and lower than the collusive ones. Specifically, we set $A_i = \left\{70, 72\frac{1}{2}, ..., 102\frac{1}{2}, 105\right\}$, so that $v = 2\frac{1}{2}$ and $m = 8$. Thus, the price may range from $290 - 105 \times 2 = 80$ to $310 - 70 \times 2 = 170$, with 37 possible levels in total.

With perfect monitoring, the set of states is $S = A \times A$ and thus the Q-matrix has $15^3 = 3,375$ entries, exactly as in the Bertrand model of Calvano et al. (2020). To facilitate comparisons with that paper, we use the same learning and experimentation parameters. Thus, we focus on the case where $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$.[16]

With imperfect monitoring, the state space coincides with the set of possible prices, which is $S = \left\{80, 82\frac{1}{2}, ..., 167\frac{1}{2}, 170\right\}$, so the Q-matrix has $15 \times 37 = 555$ entries. The fraction of price levels that do not fully reveal the rivals' output is roughly a third. Since there are fewer states, any given value of β now effectively entails more experimentation (i.e., each cell of the matrix is visited more often by random exploration).

4. RESULTS

Each set of parameter values defines an "experiment," for which the evolution of play can be calculated numerically. The evolution is stochastic, however, as demand is subject to random shocks, and both the actions (when the algorithms are in exploration mode) and the initial state are drawn randomly. To smooth out uncertainty, for each experiment we run 1,000 sessions and then average the results across sessions.

In every session, an algorithm repeatedly plays against an identical opponent. The session continues until the algorithms' behavior stabilizes. We take that to mean that the optimal strategy for each player does not change for 100,000 consecutive periods. In spite of the lack of any theoretical guarantee, in our simulations the algorithms always settled to a stable behavior – a *limit strategy*. However, the learning process is slow, and convergence

[16]We refer to Calvano et al. (2020) for a discussion of the choice of the learning and experimentation parameters. Among the extensions, we have considered a grid of 100×100 values of α and β, finding that the results are robust to the choice of α and β in a reasonable range.

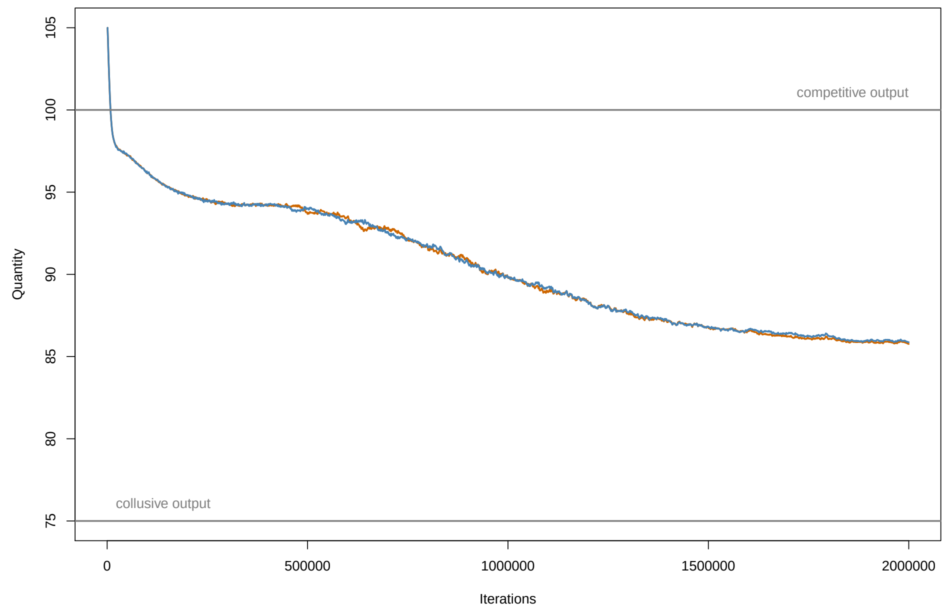

Figure 1: Evolution of greedy output levels of the two algorithms. The greedy output level is the one associated with the highest Q-value for the current state. The output level actually chosen may be different due to experimentation.

to a limit strategy typically takes a large number of repetitions, in the order of hundreds of thousands.

4.1. *Outputs and profits*

Figure 1 shows the evolution of each firm's output. Even though collusion is far from perfect, it is evident that the algorithms manage to significantly contract their output levels. Output falls below the competitive level quite early and keeps declining as the learning process continues.

The contraction of output translates into a substantial increase in profits (Figure 2). To facilitate comparisons, we use a normalized measure of the profit change:

$$\Delta \equiv \frac{\overline{\pi} - \pi^C}{\pi^M - \pi^C}, \tag{6}$$

where $\overline{\pi}$ is the average per-firm profit upon convergence. Thus, $\Delta = 0$ corresponds to the non-cooperative outcome and $\Delta = 1$ to the perfectly collusive outcome. In our baseline experiment, once the algorithms have converged to a limit strategy and hence the learning

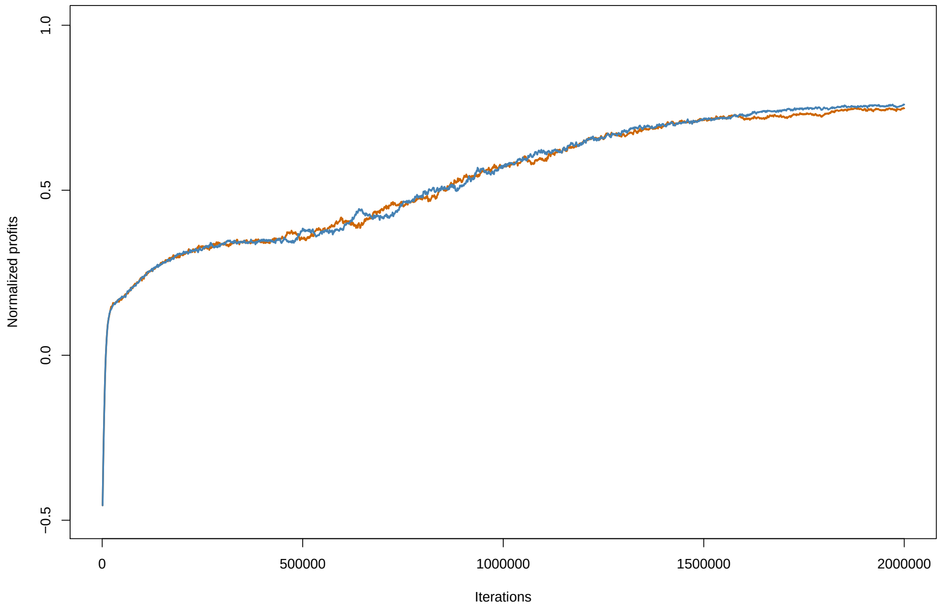

Figure 2: Evolution of the profit gains associated with the greedy output levels.

process is completed, the profit gain is above 75%.

4.2. *Collusive strategies*

When output is below the non-cooperative equilibrium level, profit-maximizing firms could gain, in the short run, by expanding production. Our algorithms evidently refrain from doing so, but this may be either because they fail to optimize, or because they have come to a tacit agreement that any output expansion would be punished in the subsequent periods, making the move unprofitable. It is only in this latter case that we have genuine collusion, so it is important to ascertain whether our algorithms have learned to punish deviations, and if so, how.

In principle, punishments might take different forms. For example, Green and Porter (1984) focus on strategies where firms stick to a low, collusive output level as long as the price stays above a certain threshold but set an higher output, entering into a "price war," when the price falls below the threshold. The price war is temporary, though, so after some time cooperation is resumed. This latter property is essential in stochastic environments, where the standard "grim trigger" strategies would imply that if the market is hit by an adverse demand shock, firms would be trapped in an infinitely long punishment phase. (With reinforcement-learning, algorithms engage in active experimentation and thus the punishment could also be triggered by the algorithms' exploration.)

However, Green and Porter's strategies cannot be implemented with one-period memory. The reason for this is that when the algorithms execute the punishment, the price remains below the threshold and thus the punishment is repeated also in the next period, and so on forever. The problem is, algorithms who recall only the last-period events do not possess a clock that tells them when it is time to end the punishment and re-start to cooperate. Thus, if our algorithms have learned to punish deviations, they must have learned something different from the Green and Porter strategies.

To verify what form of punishment our algorithms have learned, we consider the limit strategies that underpin the non-competitive outcomes described above. In our setting with a one-period memory, strategies are functions that map period-$t-1$ price into period-t output. As it turns out, there is considerable variation in the limit strategies that the algorithms converge to. These limit strategies depend quite sensitively on the specific history of the interaction between each pair of algorithms, which is noisy because the algorithms experiment extensively in a random fashion.

Averaging across the 1,000 sessions of the baseline experiment, however, eliminates much of the noise and reveals a clear pattern. Figure 3 depicts the average limit strategy which the algorithms converge to. To facilitate the interpretation of the strategy, the figure also depicts the demand functions in the high- and low-demand states, re-scaled at the firm level, allowing one to directly map the per-firm output into the equilibrium price.

In spite of the algorithms' short-lived memory, the average strategy is remarkably similar in spirit to those of Green and Porter (1984). Over the relevant range, a firm's output is a smooth decreasing function of the past price. This implies that an algorithm that observes a fall in the market price reacts by increasing its output (the punishment). However, the curve that represents the average limit strategy is flatter than the demand schedules, implying that the punishment is mild enough to result in a price increase relative to the previous period. In turn, this implies that the next-period punishment will be milder. Period after period, the punishment fades away, and the market gradually returns to its resting point (i.e., the collusive outcome), where the punishment eventually ends. In this way, the intensity of the punishment acts as a surrogate of the clock, allowing the algorithms to implement punishments of finite duration. With strategies of this sort, collusive outcomes can be sustained in stochastic environments even with a one-period memory.

The two panels in Figure 4 illustrate more directly the punishments built in the algorithms'

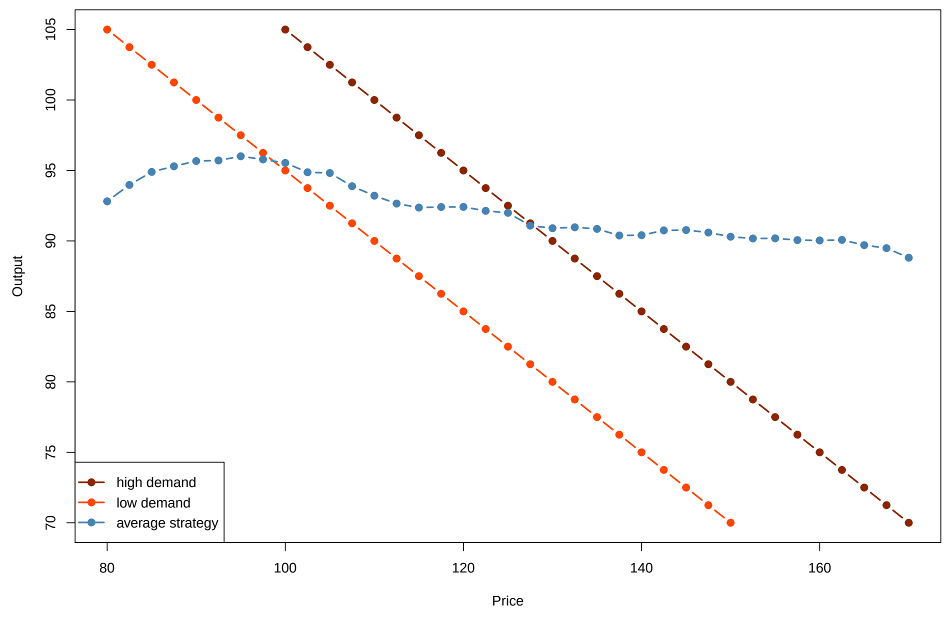

Figure 3: The average limit strategy. The two straight lines are the demand curves in the high- and low-demand state, re-scaled down at the firm level.

limit strategies. They show an algorithm's response to an exogenous output expansion by the rival. That is, starting from the outputs the algorithms have converged to, we exogenously force one algorithm to defect by expanding production. The other algorithm instead continues to play according to his learned strategy. We then examine the reaction of the algorithms in the subsequent periods, when the forced cheater reverts to his learned strategy as well. To avoid confounding effects, during this process demand is freezed and always remains either in the low or the high state. Figure 4 shows that deviations get punished, but after some time the algorithms gradually return to their pre-deviation behavior. Naturally, the punishment is softer when demand is high, as in this case a deviation may be confounded with a negative demand shock. Starting from the low-demand state, in contrast, there is no risk of confusion, and hence the punishment is harsher.[17]

Naturally, the limit strategy portrayed in Figure 3 implies that the algorithms execute a punishment even after an adverse demand shock. As argued by Green and Porter (1984), this is necessary for otherwise a smart player could deviate by hiding behind the demand

[17]The fact that the algorithms' punishments are not very harsh may be reminiscent of Bernheim and Madsen (2017). Their analysis, however, focuses on mixed strategies, whereas our algorithms (when in exploitation mode) play pure strategies by design (ties are broken by chosing the higher output).

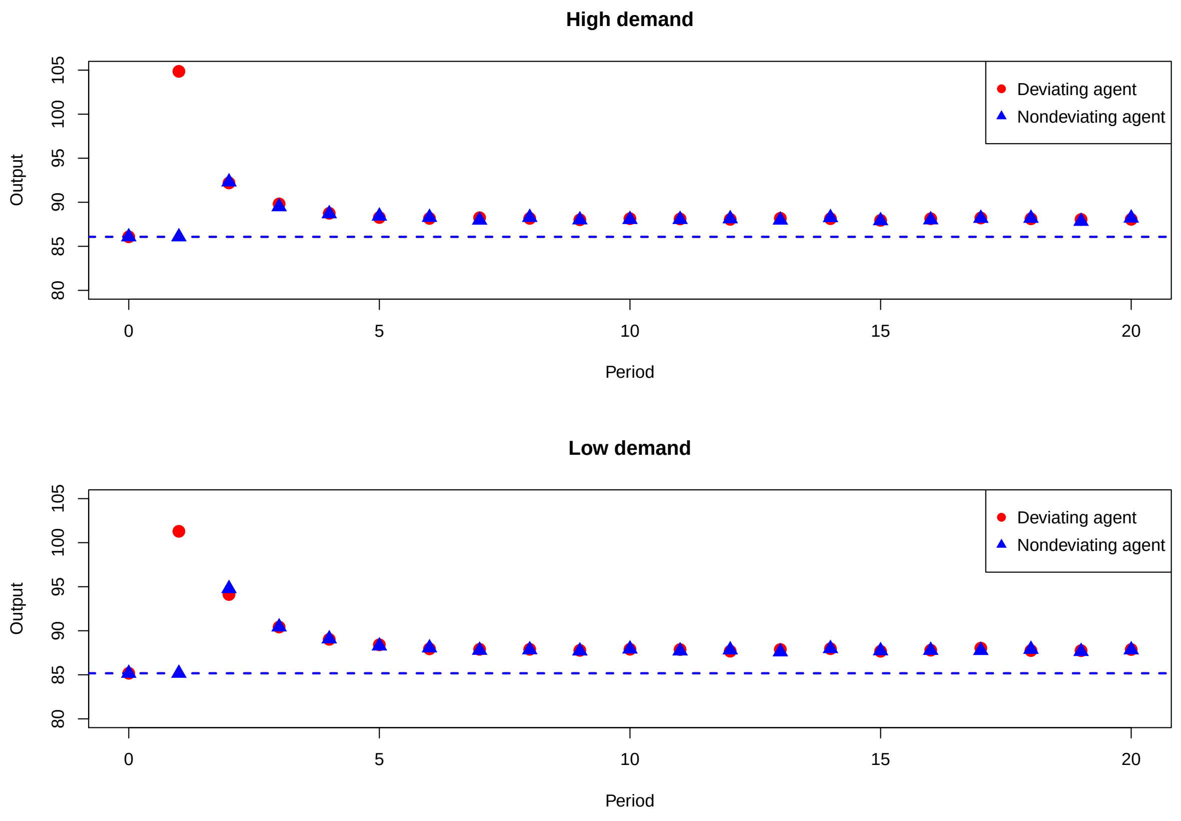

Figure 4: The evolution of output levels following an exogenous deviation by one of the two algorithms, starting from the high- and low-demand state (respectively, upper and lower panel). The figure represents the average over 1,000 sessions.

	Deterministic Demand	Stochastic Demand
Perfect Monitoring	84.16 %	79.72%
Imperfect Monitoring	89.60%	76.25%

ABLE ITABLETABLE I
The impact of imperfect monitoring

shocks, thereby avoiding the punishment. Since price wars must occur not only off but also on the equilibrium path, imperfect monitoring inevitably hinders collusion. We next ask to what extent it does.

4.3. *The impact of imperfect monitoring on collusion*

The fact that the algorithms are still able to autonomously learn genuinely collusive behavior does not mean that imperfect monitoring has no impact on collusion. Collusion may be more or less complete, and imperfect monitoring may hinder algorithmic collusion to some extent.

To assess the impact of imperfect monitoring on collusion, it may be useful to consider some relevant benchmarks, for each of which we have conducted numerical simulations in the same economic environment described above (see Table 1). The first benchmark is one of perfect monitoring and no uncertainty.[18] The profit gain here is 84.16%, about 8% higher than in our baseline experiment.

The difference between these values, however, can be due to the imperfectness of monitoring or the uncertainty of demand in itself. To disentangle these two effects, we consider two more benchmarks. In one, we still have perfect monitoring but now with demand uncertainty. Like in Calvano et al. (2020), uncertainty reduces the profit gain. Here, the effect is on the order of 5%.

The next and last benchmark is a hybrid model with no uncertainty and "imperfect monitoring," in the sense that the algorithms may condition their current outputs only on the past price level. With no uncertainty the price level is fully revealing, but the different specification of the state space may change the way the algorithms learn and

[18]This setting compares directly with our previous work on Betrand competition, as the size of the Q-matrix, and hence the effective level of experimentation entailed by the chosen value of β, is the same. As it turns out, the average profit gain upon convergence, which was 84.9% in Calvano et al. (2020), is almost identical.

eventually behave, compared to the case of perfect monitoring.[19] Indeed, the profit gain is 89.6% and hence is higher than in the corresponding case with perfect monitoring. There are two possible explanations for this result. First, when the state space is the set of possible prices the Q-matrix is smaller, and hence the same level of the parameter β translates into wider experimentation, possibly allowing for better learning.[20] Second, with fewer states the algorithms' strategies are simpler, and this may help them to better coordinate.

With these benchmarks at hand, we can apply differences-in-differences to control for the effect of demand uncertainty, identifying the effect of imperfect monitoring. This is the difference between the cases of imperfect monitoring with and without uncertainty, i.e. $(76.25 - 89.60)$, minus the analogous difference under perfect monitoring, i.e. $(79.72 - 84.16)$. Thus, imperfect monitoring reduces profit gains by $13.35 - 4.44 = 8.91\%$ in absolute terms. The effect is appreciable, but not very large.[21]

5. ROBUSTNESS

Starting from the baseline experiment, we considered a number of extensions. In this section, we briefly report the main results of this robustness analysis.[22]

5.1. *Learning and experimentation parameters*

We varied the learning and experimentation hyper-parameters considering the same grid of 100×100 values of α and β as in Calvano et al. (2020). We find that over the entire grid, the value of the profit gain ranges from 65% to over 80%. This confirms that collusive outcomes are common, not obtained at just a few selected points of the grid. In particular, the profit gain tends to decrease with the experimentation parameter β, that is, to increase with the extent of experimentation. This higher profit gain comes at the cost of longer times to convergence, though.[23]

[19]This hybrid case is the one considered by Abada and Lambin (2020) in their analysis of the electricity market.

[20]This is not a foregone conclusion, though, as both firms experiment in a symmetric fashion, and the rival's experimentation creates noise that may impede a firm's learning.

[21]This is consistent with the theoretical literature, which shows that imperfect monitoring hinders collusion with fully rational players.

[22]The detailed results, as well as the code used for the simulations, can be obtained from the authors upon request.

[23]Conversely, increasing the exploration parameter may reduce the time to convergence to on fifth of the baseline case. This is an instance of the trade-off faced by the algorithms' designers when setting the hyper parameters.

5.2. *Number of firms*

Results are robust to an increase in the number of firms. Consistently with theory, collusion is more difficult when there are more players. However, with 3 firms the average profit gain is 72.7%, not much lower than in a duopoly; with 4 firms, it is still 66.6%.

5.3. *Longer memory*

We allowed firms to condition their period-t output not only p_{t-1} but also on p_{t-2} and p_{t-3}. With a 2-period memory, the average profit gain is almost the same as in the baseline model (75.7%). A 3-period memory reduces the profit gain to 64.5%. This result may seem surprising, as a longer memory allows for more flexible strategies. We conjecture that it is due to the fact that as memory increases the Q-matrix becomes bigger, hindering learning. Consistently with this conjecture, the time required to complete the learning is significantly longer with 2 and especially 3-period memory.

5.4. *Correlated shocks*

We allowed demand shocks to be correlated across time. With a coefficient of auto-correlation of 0.9, the average profit gain is 86.7 %.

5.5. *Off-line training*

As noted, our algorithms takes a very large number of periods to stabilize their behavior. That learning 'on the job' can be lengthy and costly is well known in the computer science literature. A standard approach to deal with this, is to train the algorithms in synthetic environments (i.e. 'off-line') before putting them to work.[24] If at least part of learning can be carried out off-line, then the algorithms might start to collude much earlier than the above analysis suggests. But since the way the algorithms learn to collude is rather idiosyncratic, depending on the specific history of experimentation experienced in each individual simulation, it is not clear to what extent the knowledge gained in off-line training may be useful to coordinate with a different player.

To shed more light on this issue, we re-match the algorithms once they have converged and let them start to play again with different opponents. In the newly formed pairs, we

[24]For example, the celebrated chess program *Alpha Zero* played billions of matches against a clone of itself before being deployed in tournaments against humans or other artificial players.

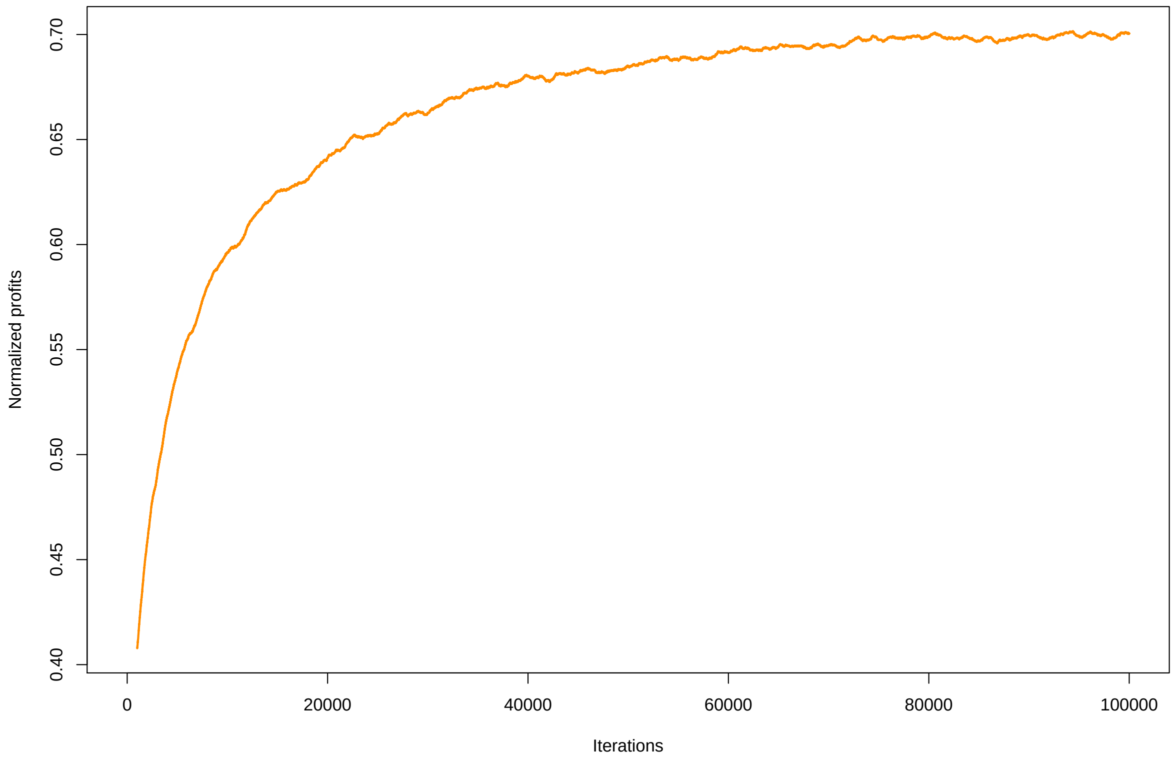

Figure 5: Evolution of the average profit gain when the algorithms are re-matched upon convergence (average across 1,000 re-matching exercises).

shut exploration down by setting $\varepsilon = 0$. Faced with the "unexpected" choices made by the new competitor, initially the algorithms keep trying actions that performed well in the past. As they discover that these choices are no longer good in the new environment, the algorithms update their Q-matrixes and hence change their strategies. After a learning phase, however, they once again stabilize their behavior.

Figure 5 shows the evolution of the average profit gain for such re-matched pairs. Initially, the profit gain falls from 75% to about 40%, confirming that coordination is largely pair-specific. However, the profit gain rises quite rapidly as the algorithms adapt to the new environment, exceeding 60% in a few thousands repetitions and approaching the new long-run level in about 5-10% of the time it originally took to converge.

	Deterministic Demand	Stochastic Demand
Perfect Monitoring	84.16 %	84.42%
Imperfect Monitoring	89.60%	71.93%

ABLE IITHE IMPACT OF IMPERFECT MONITORING WITH GREATER UNCERTAINTY.TABLETABLE II

The impact of imperfect monitoring with greater uncertainty.

5.6. *Higher uncertainty*

The small size of the effect of imperfect monitoring on the level of collusion identified above could be due to the limited amount of uncertainty that we have assumed in the baseline experiment, where the demand shock is smaller than 10% in relative terms. As a final robustness check, we have therefore examined how the impact varies if uncertainty increases.

As a preliminary step, we have enlarged the action space by allowing each firm to choose among $k = 27$ equally spaced output levels from 60 to 125, so that the step size v remains $2\frac{1}{2}$ as in the baseline specification. This leads to a profit gain of about 80% which, if anything, is slightly larger than the baseline profit gain.

Next, we have considered $h = 5$ equiprobable demand levels, ranging from $d^1 = 250$ to $d^5 = 350$. Thus, a shock can be one third of the average demand. The value of d is distributed identically and independently across periods. The absence of auto-correlation among the shocks implies that our algorithms are now facing considerable uncertainty. Nevertheless, the profit gain is still a sizeable 72%. The corresponding level of the profit gain with perfect monitoring is 84%. Applying again differences-in-differences (see Table 2), one sees that the impact of the imperfectness of monitoring is now about 17%.

Figure 6 shows the average limit strategy for this case. For relatively high prices, the curve is almost flat, meaning that over this range price changes do not trigger any punishment. For lower prices, however, the curve is slightly decreasing. Thus, large deviations and/or big negative demand shocks entail temporary punishments, qualitatively similar to those described for the baseline experiment.

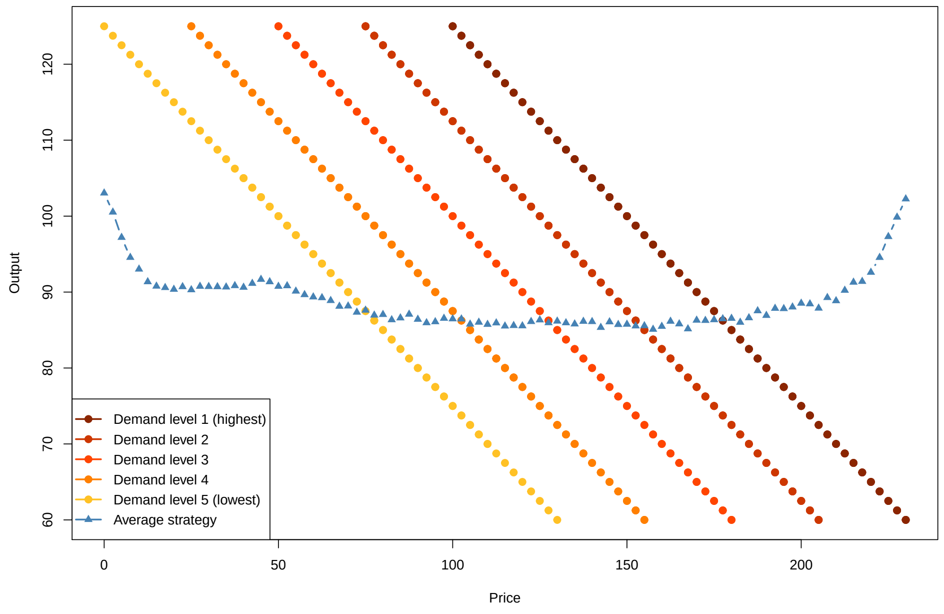

Figure 6: The average limit strategy with greater uncertainty. The decreasing straight lines are the firm-level demand curves in the five states of demand.

6. CONCLUSION

We have conducted numerical simulations in synthetic environments where identical Q-learning algorithms interact for a large number of periods. These environments are constructed from a model of Cournot competition with imperfect monitoring adapted from Green and Porter (1984). In the perfect monitoring benchmark, the behavior of the algorithms is remarkably similar to that displayed in the Bertrand setting of our previous work (Calvano et al. 2020). With imperfect monitoring, the level of collusion decreases. The effect is appreciable but not very large, meaning that the algorithms are capable of colluding even under imperfect monitoring.

The strategies that sustain the collusive outcomes are quite sophisticated: the algorithms enter into a “price war” both after a deviation by the rival and a demand shock. The price war is temporary, however, and lasts for some periods, after which cooperation re-starts. The algorithms are able to implement these strategies even if they cannot communicate and they are endowed with a short-lived memory, as they learn to use the intensity of the punishment as a surrogate of a longer memory.

These findings indicate that imperfect monitoring is not an insurmountable obstacle to autonomous algorithmic collusion. Like previous work in this area, however, this paper showed the possibility of autonomous algorithmic collusion only in basic setups. To assess the likelihood of algorithmic collusion, future work must focus on diverse and asymmetric algorithms, the speed of learning, the possibility of firm-specific shocks, and other factors that may hinder algorithmic collusion.

REFERENCES

Abada, I. and Lambin, X. (2020). Artificial Intelligence: Can seemingly collusive otcomes be avoided? https://papers.ssrn.com/abstract=3559308

Abreu, D., Pearce, D., and Stacchetti, E. (1986). Optimal cartel equilibria with imperfect monitoring. Journal of Economic Theory, 39(1), 251-269.

Abreu, D., Pearce, D., and Stacchetti, E. (1990). Toward a theory of discounted repeated games with imperfect monitoring. Econometrica: Journal of the Econometric Society, 1041-1063.

Assad, S., Clark, R., Ershov, D., and Xu, L. (2020). Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market. https://papers.ssrn.com/abstract=3682021

Back, K., Cao, C. H., and Willard, G. A. (2000). Imperfect Competition among Informed Traders. The Journal of Finance, 55(5), 2117-2155.

Bernheim, B. Douglas, and Erik Madsen. (2017). Price Cutting and Business Stealing in Imperfect Cartels. American Economic Review, 107 (2): 387-424.

Calvano, E., Calzolari, G., Denicolò, V., Harrington, J. E., and Pastorello, S. (2020). Protecting consumers from collusive prices due to AI. Science, 370(6520), 1040-1042.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. The American Economic Review, 110(10), 3267-3297.

Colla, P., Distaso, W., and Vitale, P. (2020). A Model of Price Manipulation and Collusion around the London 4pm Fix. *Mimeo.*

Foster, F. D., and Viswanathan, S. (1996). Strategic Trading When Agents Forecast the Forecasts of Others. The Journal of Finance, 51(4), 1437-1478.

Green, E. J., and Porter, R. H. (1984). Noncooperative Collusion under Imperfect Price Information. Econometrica: Journal of the Econometric Society, 52(1), 87-100.

Johnson, J., Rhodes, A., and Wildenbeest, M. R. (2020). Platform Design When Sellers Use Pricing Algorithms. In Available at SSRN. https://doi.org/10.2139/ssrn.3691621

Klein, T. (2019). Assessing Autonomous Algorithmic Collusion: Q-Learning Under Short-Run Price Commitments. https://doi.org/10.2139/ssrn.3195812

Maskin, E., and Tirole, J. (1988). A Theory of Dynamic Oligopoly, II: Price Competition, Kinked Demand Curves, and Edgeworth Cycles. Econometrica, 56(3), 571-599.

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

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