ALGORITHMIC COLLUSION WITH IMPERFECT MONITORING[†]

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

November 2020

We show that 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

Firms are increasingly delegating their strategic choices to algorithms of various types, including some that are capable of learning autonomously by engaging in active experimentation. Concerns have been voiced that if these algorithms are adopted by would-be competitors, they may learn to collude without having been specifically instructed to do so and without communicating with one another. This possibility would pose a challenge for antitrust policy because in most jurisdictions the illegality of collusion today resides in the *communications* among the firms and their *conscious* adoption of collusive rules of conduct. Since the algorithms do not communicate and are not conscious, algorithmic collusion has the potential to defy current policy (see Calvano, Calzolari, Denicolò, Harrington and Pastorello (2020)).

Corresponding author: Vincenzo Denicolò, vincenzo.denicolo@unibo.it

[*]Bologna University; [‡]Toulouse School of Economics; [&]European University Institute; [§]CEPR

Whether autonomous algorithmic collusion is a real risk, however, is still a matter of debate. Two avenues have been followed to assess the risk. 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 pricing algorithms is associated with a sizeable increase in the stations' margins. Another approach studies the behavior of algorithms under controlled conditions in simulated markets. For example, Klein (2019) conducts this kind of analysis for the Maskin and Tirole (1988) model of staggered pricing, and our previous work considers an infinitely repeated Bertrand game where firms can perfectly observe rivals' prices.[1] Both studies find that algorithmic collusion arises spontaneously in such environments.

This paper contributes to the latter strand of research by analyzing the case of imperfect monitoring. To this end, we use the classic model of Green and Porter (1984), where firms set quantities and observe the price level but cannot perfectly infer rivals' outputs because demand is stochastic.[2]

The case of imperfect monitoring is interesting for at least three reasons. First, algorithms are increasingly used not only in marketplaces such as Amazon, where each seller can easily monitor rivals' prices in real time, but also in markets where rivals' strategies are not easy to observe. Examples include financial markets with agents possessing inside information,[3] and the intraday markets for electricity.[4] Models of Cournot competition with imperfect monitoring may provide a good fit for these markets. Second, the imperfectness of monitoring poses a tougher challenge for the algorithms, so the analysis of this case may help assess, more generally, to what extent algorithms are capable of colluding in complex environments. Third, the analysis may be interesting for policy purposes, as one possible route to avoid algorithmic collusion is to suppress some available information, for instance about rivals' prices or outputs. The present paper may shed light on the effectiveness of this regulatory approach.

We find that Q-learning algorithms learn to collude even with imperfect monitoring. Naturally, the level of collusion decreases as demand shocks get bigger. However, the effect seems limited and would arise even with fully rational (human) agents.

[1] See Calvano, Calzolari, Denicolò, and Pastorello (2020) – henceforth Calvano et al. (2020). See also Johnson, Rodhes and Wildenbeest (2020).

[2] This model is well understood and has become the workhorse of the theory of repeated games with imperfect monitoring: see for instance Abreu, Peace and Stacchetti (1986) and (1990).

[3] See for instance Foster and Viswanathan (1996) and Back, Cao and Willard (2000). For an application to collusion in financial markets, see Colla, Distaso and Vitale (2010).

[4] See, for instance, Aid, Gruet and Pham (2016).

The strategies that the algorithms learn are remarkably similar to those considered by Green and Porter. 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, however, is that the rational agents of Green and Porter possess an infinitely long memory and thus know when to end the punishment. Our algorithms, in contrast, have a short-lived memory, which impedes the implementation of punishments of finite duration. To circumvent this problem, they resort to an ingenious method that will be described in greater detail below.

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 reports the results of the analysis. Section 5 offers some concluding remarks.

2. Q-LEARNING

To facilitate comparison with previous work, we focus on Q-learning algorithms. In this section, we shall first provide a short, abstract description of these algorithms and then specify how they are implemented in our setting.[5]

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 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

[5]We refer the reader to Calvano et al. (2020) for a self-contained introduction to Q-learning, and to Sutton and Barto (2018) for a more in-depth treatment. Other analyses of collusion among Q-learning algorithms include Waltman and Kaymak (2008) and Klein (2019).

Q-function $Q_i(s, a_i)$, which represents the cumulative discounted payoff of taking action 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 the Q-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 $a_{i,t}$ in state s_t, the algorithm observes π_t 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:

$$Q_{it+1}(s, a_i) = (1 - \alpha)Q_{it}(s, a_i) + \alpha \left[\pi_{i,t} + \delta \max_{a_i \in A_i} Q_{it}(s', a_i)\right]. \tag{3}$$

For all other cells $s \neq s_t$ and $a_i \neq a_{it}$, the Q-value does not change: $Q_{it+1}(s, a_i) = Q_{it}(s, a_i)$. 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 the complementary 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-declining exploration rate:

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

where $\beta > 0$ is a parameter. This implies that initially the algorithms choose in purely random fashion,[6] 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 demand parameter d_t is subject to random shocks. In particular, we shall focus on the case of duopoly, $n = 2$. The symmetric, constant marginal costs are normalized to zero. Thus, the per-period reward accruing to firm i is $\pi_{it} = p_t q_{it}$.

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. The linearity of demand implies that uncertainty does not affect equilibrium outputs and profits as long as the expected value of d stays constant.[7]

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 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

[6]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.

[7]This follows fom the fact that output is chosen before demand is realized.

therefore condition its current choice on them. Assuming a one-period memory,[8] the 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 $s_{it} = \{q_{it-1}, p_{t-1}\}$.[9] 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 shall focus on strategies of this type, where a state is simply the past 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.[10] The extent to which this may happen, and hence monitoring be imperfect, can be indexed by the fraction of possible price levels that do not fully reveal the rival's output. With our assumptions, this is:

$$\mathcal{O} = 1 - \frac{m}{k} \tag{6}$$

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.

[8] A finite memory guarantees that the state space can be finite and time invariant, allowing to implement Q-learning in our economic framework.

[9] Note that in this case the set of states is firm specific.

[10] 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.

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 the comparison with the Bertrand case, we use the same learning and experimentation parameters as in our previous paper. In particular, we conduct our simulations for the same grid of 100×100 values of α and β,[11] and in the exposition of the results we focus on the same point of the grid, i.e. $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$.[12]

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. 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 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 the same 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 to a limit strategy typically takes a large number of repetitions. We refer the reader to our previous work, Calvano et al. (2020), for a discussion of the issue of the speed of learning.

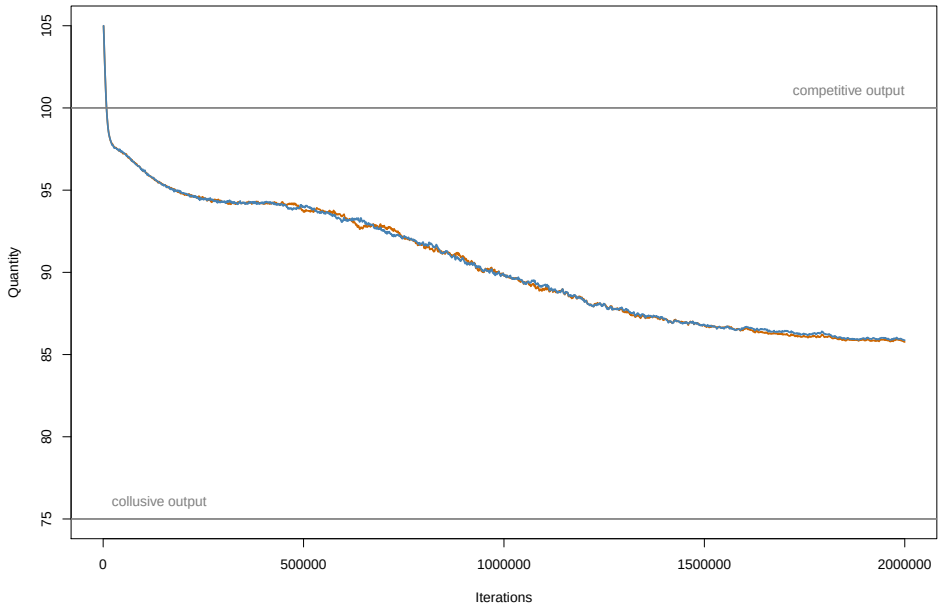

Figure 1: The diagram represents the greedy output levels of the two algorithms abstracting from experimentation

4.1. *Outputs and profits*

Figure 1 shows the evolution of individual outputs. Even though collusion is not perfect, it is evident that the algorithms manage to appreciably 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:

$$\Delta \equiv \frac{\overline{\pi} - \pi^C}{\pi^M - \pi^C}, \tag{7}$$

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

[11]The results for the entire grid of parameter values are available from the authors upon request.

[12]We refer to Calvano et al. (2020) for a discussion of the choice of the learning and experimentation parameters.

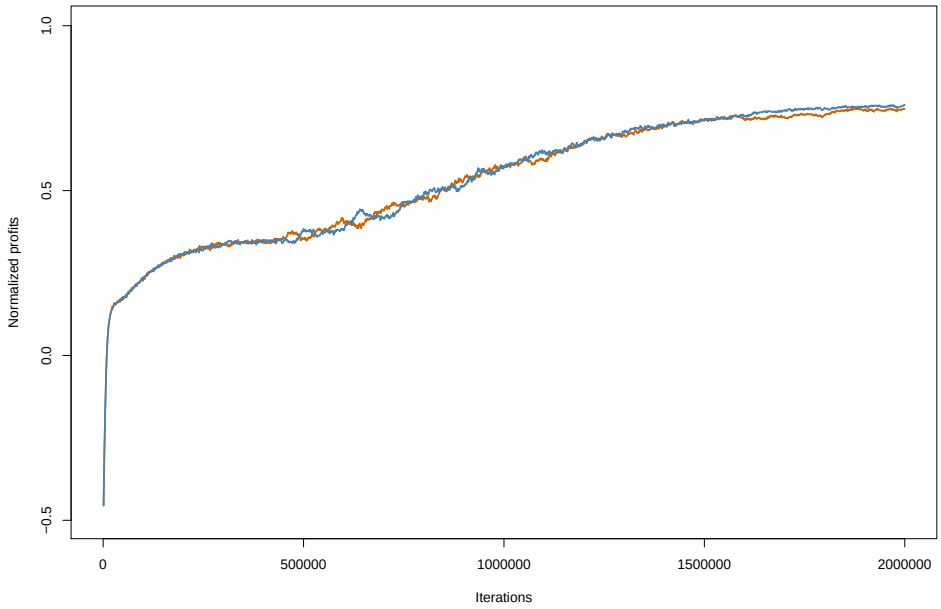

Figure 2: The evolution of the profit gains.

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 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 that engage in active experimentation, the punishment could be triggered also by the algorithms' exploration.)

However, Green and Porter's strategies cannot be implemented with a memory of one

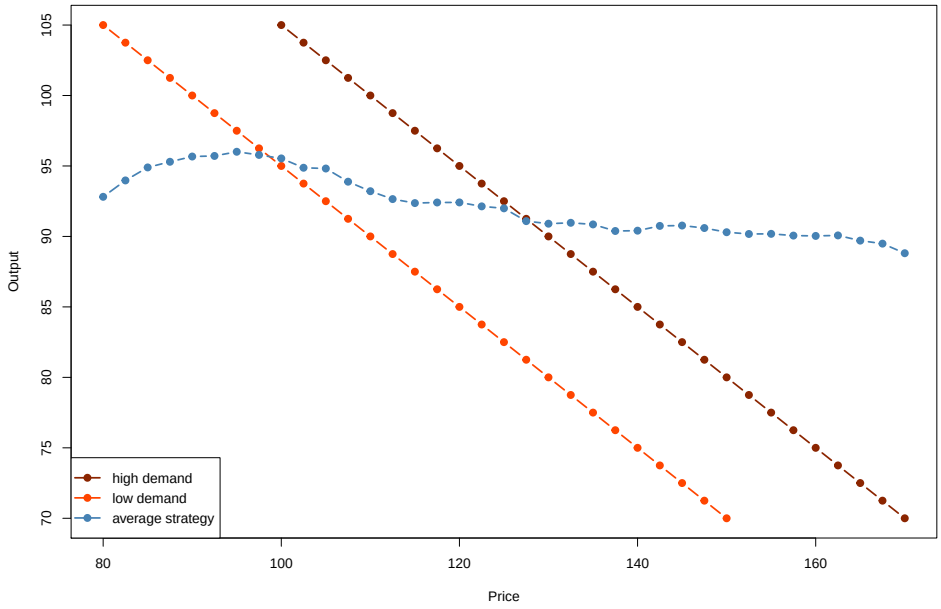

Figure 3: The average limit strategy. The two straight lines are the demand curves in the high- and low-demand state, rescaled down at the firm level.

period only. 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.

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 the period-$t-1$ price into the period-t output. Figure 3 represents the average limit strategy in the baseline experiment, along with the demand functions in the high- and low-demand states, respectively. (The demand functions are re-scaled at the firm level, allowing one to directly map the per-firm output into the equilibrium price.)

Figure 3 shows how our algorithms have come to implement strategies that are similar in spirit to those of Green and Porter in spite of their short-lived memory. Over the relevant range, a firm's output is a smooth decreasing function of the past price. This implies that when a deviation by the rival (or a negative demand shock) causes a fall in the price, the firm reacts by increasing its output, thereby punishing the deviation. Crucially, however,

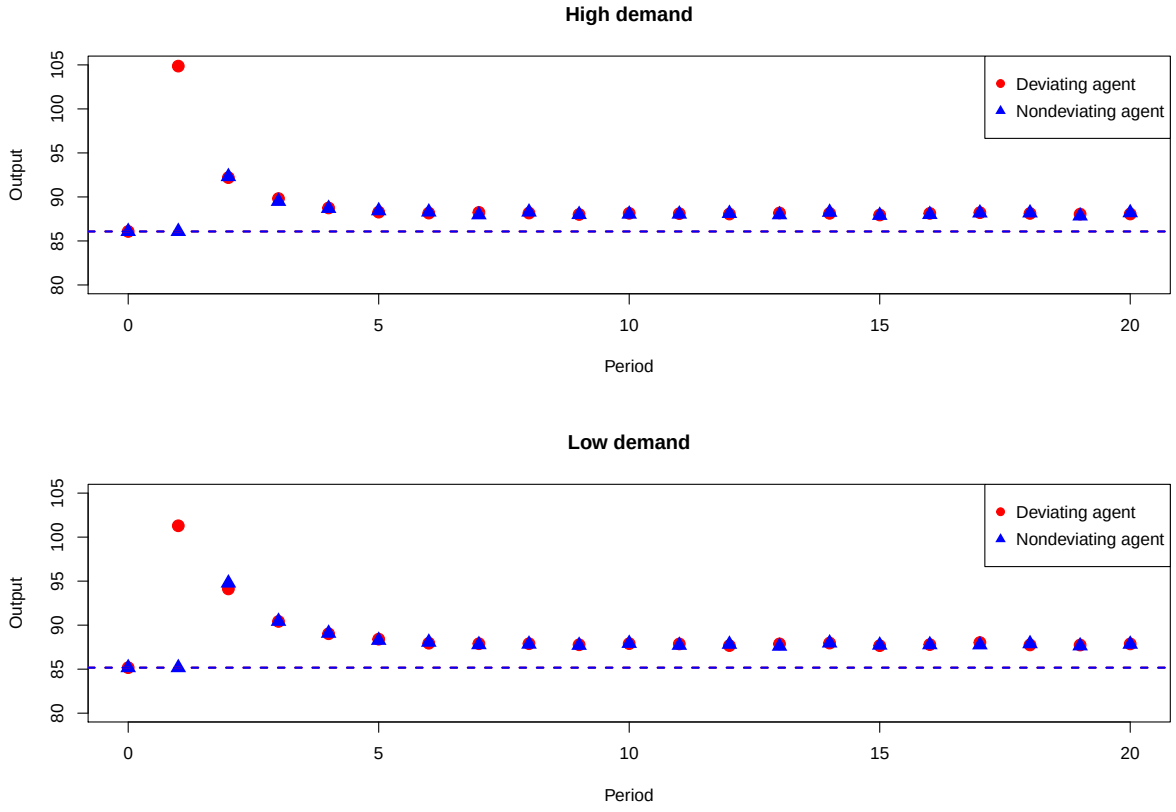

Figure 4: The evolution of output levels following an exogenous deviation by one of the two algorithms, starting from the high- and low-demand state (respectively, upper and lower panel).

the intensity of the punishment is an increasing function of the fall in the price. This implies that if, after the initial punishment, the price remains below the collusive level but increases a bit, in the subsequent period the punishment will continue but will be less intense. Period after period, the punishment becomes milder, and the market gradually returns to its resting point (i.e., the collusive outcome), where the punishment eventually ends. Thus, 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' 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

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

TABLE I
THE IMPACT OF IMPERFECT MONITORING

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.

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, this is necessary for otherwise a smart player could deviate by hiding behind the demand 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. The profit gain here is 84.16%, about 8% higher than in our baseline experiment.[13]

[13]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.

The difference between these profit gains, however, can be due to two different reasons: the imperfectness of monitoring, and the uncertainty of demand. To disentangle the 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 algorithms' 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 eventually behave, compared to the case of perfect monitoring. 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, with "imperfect monitoring" the Q-matrix is smaller, and hence the same level of the parameter β translates into wider experimentation, possibly allowing for better learning.[14] Second, with fewer states the algorithms' strategies are simpler, and this may help them to better coordinate.[15]

With these benchmarks at hand, one can apply differences-in-differences to identify the net effect of the imperfectness of 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. $(79.72 - 84.16)$. It therefore appears that the imperfectness of monitoring in itself reduces the profit gain by $13.35 - 4.44 = 8.91\%$. The effect is appreciable, but for the baseline experiment it is not very large.[16]

4.4. *Robustness*

The small size of the effect 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 robustness check, we have therefore examined how the impact varies if uncertainty increases.

As it turns out, the average profit gain upon convergence, which was 84.9% in Calvano et al. (2020), is almost identical.

[14]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.

[15]As a robustness check, we have considered also the case in which demand shocks are autocorrelated, with a coefficient of correlation of 90%. The profit gain with imperfect monitoring is about 86%, not much lower than with no shocks. Note that the level of persistency considered is actually rather low, as algorithms are typically adopted when decisions can be changed very frequently, and hence periods can be very short.

[16]Remember that even fully rational players could not collude perfectly in the presence of imperfect monitoring.

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

TABLE II

The impact of imperfect monitoring with greater uncertainty

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 autocorrelation 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%.

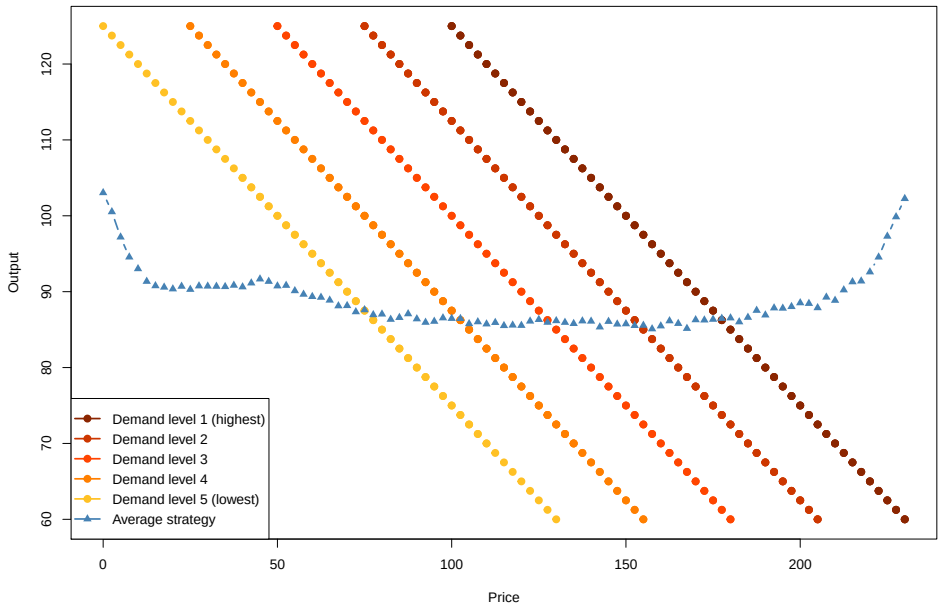

Figure 5: The average limit strategy with greater uncertainty. The decreasing straight lines are the firm-level demand curves in the five states of demand.

Figure 5 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.

5. CONCLUSION

We have analyzed the behavior of Q-learning algorithm in a model of Cournot competition with imperfect monitoring adapted from Green and Porter (1984). In the perfect monitoring benchmark, the behavior of our 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. 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 several periods, after which cooperation restarts. 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.

REFERENCES

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.

Aïd, R., Gruet, P., and Pham, H. (2016). An optimal trading problem in intraday electricity markets. Mathematics and Financial Economics, 10(1), 49-85.

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.

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, Distaso and Vitale (2010) 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.