Exploring Robustness of Automated Pricing Algorithmic Collusion in Financial Markets

Ashish Rana, 1822317
ashish.rana@students.uni-mannheim.de

July 7, 2023

Contents

List of Tables

Abstract

In the modern digital economy automatically setting commodity prices for dynamically varying demands plays a huge role for maximizing the profits for an organization. Often this helps the organizations to gain advantages over their competitors, and increase their survival rate in the market. But, in the past it is observed that organizations often communicate and collude to maximize their profits by setting supra-competitive prices. In the age of algorithmic pricing systems this collusion is possible without communication as well with the help of modern reinforcement learning (RL) algorithms. And, the current antitrust laws do not explicitly prevent such possible collusion activities across different commodity markets. With this manuscript we explore robustness of algorithmic collusion in three different RL settings, namely: i) Tabular Q-Learning, ii) Deep Q-Learning, iii) Multi-Agent Deep Q-Learning. Also, the related code repository implementation and experimentation is present at github.com/arana-initiatives/social-dilemma-collusion.

1 Introduction

With increased usage of automated pricing softwares, which holds the capacity to process data in large amounts, it is possible for firms to collude and bypass regulations. Previously, from small scale Amazon sellers to large German retail gasoline market firms have used collusion algorithms successfully to increase price-cost margins [1, 14]. This problem is only going to exacerbate in the upcoming future with an ever-evolving AI landscape. Current antitrust and competition-law targets only communication effort between firms for collusion. And, lacks appropriate regulations to avoid collusive strategies without regulations [2]. A growing literature has demonstrated the possibility of algorithmic collusion, and with this study we intend to analyze these claims in detail [3, 13].

We analyze three different studies that evaluate the same algorithmic collusion problem in different RL setups, namely: i) Tabular Q-Learning, ii) Deep Q-Learning (DQN), iii) Multi-Agent Deep Q-Learning [4, 10, 7]. As with increasing shared novelties in colluding systems, the algorithmic collusion becomes more feasible in wider oligopolies as well. With these three different setups, we analyze and quantify the robustness capabilities and recovery capacity in algorithmic collusion for agent firms. For example in the first study, even though collusion is principally feasible, agents still restrict themselves to simpler strategies, like learning uniformly from their past experiences only. Whereas in the final third study, agents supplemented with experience replay buffers give more importance to underperforming profit experiences to achieve faster supra-competitive prices with collusion. Additionally, these studies are developed from an economic perspective, therefore experiment result interpretation is the primary focus across these studies rather than result novelties. It further encourages interesting and practical discussion for this study to focus on real governance and regulations aspects as well instead of algorithmic novelties only.

In this manuscript, we first discuss the necessary background information regarding the used economic environment and reinforcement learning concepts. Second, over the span of next three sections we elaborate different collusion experiment results from the above-mentioned three studies. Third, we discuss and summarize the learnings from these studies to encourage fair and complaint competition behavior in markets with improved regulations. Finally, we conclude our exploration study and practically quantify the learnings with social dilemma based MARL experiments. [1].

2 Background Work

For simulating collusion scenarios, a repeated Bertrand competition model is chosen which formulates competing oligopolistic markets with homogenous goods. This competition assumes that goods produced are identical from consumer's perspective and the firms compete by varying prices only but not quantities. All competitors theoretically make zero profits by selling goods at marginal cost as the goods are perfect substitutes for all competing firms. Therefore, simulating this market scenario provides a perfect opportunity to analyze algorithmic collusion in abstract settings with minimal concept drift [16]. In the first subsection below, we mathematically formulate the environment to specify the dynamics of repeated Bertrand competition games. And, in the second subsection we discuss the RL problem formulation design opted by each of above mentioned studies.

[1] Experimentation code available at the repository github.com/arana-initiatives/social-dilemma-collusion

2.1 Economic Environment Description

The Bertrand competition game with two players is the baseline setup for all the experiments where players choose actions in each period t $\in$ { 0, ..., T }. In time period t for demand quantity$q_{i,t}$, the product i follows the logit demand function specified below in equation 1. In vertical differentiation a firm offers products at different price points, represented in equation 1 by γ denoting the quality parameter. Whereas, in horizontal differentiation companies differentiate themselves by offering other products or services, represented here by μ expressing goods are perfect substitutes when $\mu \to \infty$. The reward or profit at each timestep is specified by $\mathcal{R}_{i,t} = (p_{i,t}$ - $c_i)q_{i,t}$. The states S of the system is given by price profiles $(p_{1,t}, p_{2,t})$ like $S = A_1 \times A_2$, where different agent action combinations formulate the state space. This environment uses a deterministic transition function and defines agent past memory limited to problem formulation specific horizons.

$$q_{i,t} = e^{(\gamma_i - p_{i,t})/\mu} / \sum_{j=1}^{n} e^{(\gamma_j - p_{j,t})/\mu} + e^{\gamma_0/\mu} \tag{1}$$

$$\mathcal{M} = \pi - \pi^N / \pi^C - \pi^N ; \Delta_i = (\pi_i - \pi_i^N)/(\pi_i^C - \pi_i^N) \tag{2}$$

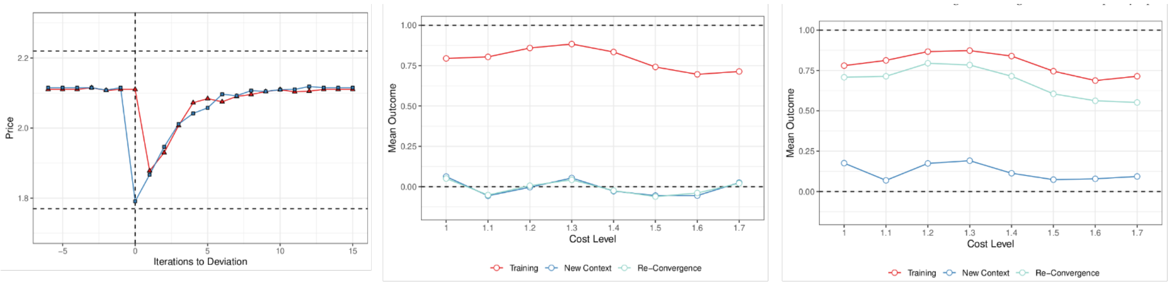
The collusion capabilities of algorithms are quantified by collusion index $\mathcal{M}$ and profit gain Δ, denoted by equations 2. By definition, collusion index is equal to averaged profit gains for two players involved in the baseline system. Here, both metrics at different granularities defines: i.) average reward in relation to static Nash equilibrium, and ii.) firm profits aimed at maximizing profits. The unique Nash equilibrium is defined over the one-shot game, where firms set their prices equal to marginal costs and make zero profit. Therefore, using Nash equilibrium as reference for these standardized metrics is important. Because both these metrics, at i^{th} firm level for Δ_i and at an average system level for $\mathcal{M}$ measures collusion tenacity. The actions are defined in a discrete manner, where i^{th} agent can choose from its action space A_i. A computationally reasonable range would be equal space prices range from the static Nash equilibrium prices $p^N_{i \in n}$ to monopoly prices $p^C_{i \in n}$ of given one-shot game. Mathematically, the action space is defined by A_i = [min($p^N_{i \in n}$) - ξ , max($p^C_{i \in n}$) + ξ], where ξ = 0.1 * [max($p^C_{i \in n}$) - min($p^N_{i \in n}$)] is the markup to increase the price range by 10 % in both directions.

2.2 Mathematical Problem Formulation

The three discussed studies formally define their collusion problem differently from theoretical perspectives while following the same centralized training and centralized execution paradigm (CTCE) paradigm [19]. The first study formulates this problem as Partially Observable Stochastic Game (POSG). As realistically it adheres to the belief that agents do not actually observe true underlying states in the market. Additionally, their formulation also uses *'context' (K)* which specifies initial seeds and reward function parameters [12]. This helps the algorithm to jointly learn collusion in specific contexts, and it further helps in assessing collusion policy extrapolation capabilities across different contexts. Whereas, the second study formulates the above economic environment as simple MDP, where the decision maker is the agent and everything else is considered as the environment. This formulation is problematic since it violates Markov property theoretically, and leaves the learning system prone to overfitting with introduced non-stationarity practically. But, to achieve faster convergence this study uses DQN with loss function formulation improvements [9]. Since, optimal joint policy cannot be represented under this MDP formulation, we get partial order of policies. The authors propose to formulate optimization objectives as a comparison of two policies, where the current policy is evaluated by the improvement it provides over the current true average reward estimate [21].

The third study also formulates the problem as MDP, and practically concatenates all agent inputs in the CTCE multi-agent system during learning. But, it exploits the temporal correlation between MDP states by using replay buffers to systematically learn from valuable past experiences [23]. The second DQN based study only uses naive replay buffers, whereas the third discussed study uses ranking mechanisms to assign more value to more relevant experiences. Practically, all three approaches use centralized executor rather than decentralized executer agents which works well in this simplistic economic environment. The state-action pair count varies across the studies to some extent based on the price range selected in each of the studies. But, still the qualitative findings are comparable since the underlying economic environment complexity is almost same across the baseline experiments. And, clearly as we keep on adding additional components like deep learning networks, and experience replay buffers, we can expect the collusion to become more robust in nature. Since, deeper and more novel RL methods can estimate the policies and value functions in a more robust manner with minimal overfitting.

3 Robust Algorithmic Collusion



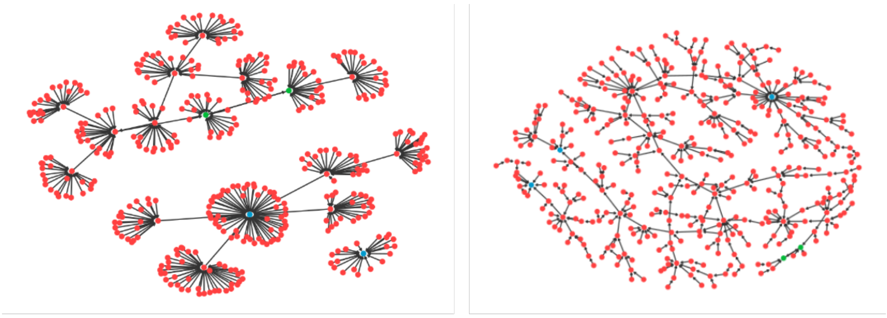
a.) Price reaction after agent defection

b.) Zero-shot new context re-convergence output

c.) Post partial training in new context re-convergence

Figure 1: Agent strategy robustness analysis in defection scenarios and differently parameterized contexts. *(Image Credit: Eschenbaum et al.)*

This foundational study carries out collusion propensity experimentation in its most nascent form with baseline Q-table based learning. But, uniquely enough this study defines the *'context'* concept, which varies in marginal cost values for tuning the experiments to different marginal cost parameters. Also, across all the training contexts with varying cost, a high collusion index between 0.70 and 0.87 is observed with highly symmetric profit gains for agents. For convergence approximately million plus timesteps are taken across all contexts on an average. This makes the collusion amongst agents highly unfeasible in realistic market setups. Experimentally it is further observed that players with high cost level lose when best response Nash play happens, and opt to reduce their prices. Vice-versa, players in low cost contexts show tendency to achieve above-Nash profits against Nash plays, which restores collusion, as highlighted in Figure 1.a. In the Figure 1.a, agent is forced for the price defection, and the second agent punishes the defecting agent with its price drop before returning to stable pre-deviation collusion prices. Further, to measure the collusion generalization capabilities this study trains the model in different context, and tests the collusion propensity in another context having different cost parameters as shown in Figure 1.b and Figure 1.c. From these Figures, we observe that if we directly evaluate the collusion models in different contexts the collusion breaks. But if we allow continued training or fine-tuning to some extent, the collusion plot again highlights reconverges back to near ideal collusion levels as earlier.



a.) Strategy pairs plot in same context

b.) Strategy pairs plot in different context

Figure 2: Agent strategy transitive closure plots for strategy evaluation in similar and different contexts. Here, the blue nodes highlight stable end-nodes with fixed action strategies, and green nodes highlight unstable end-nodes with cycles of varying action strategies. *(Image Credit: Eschenbaum et al.)*

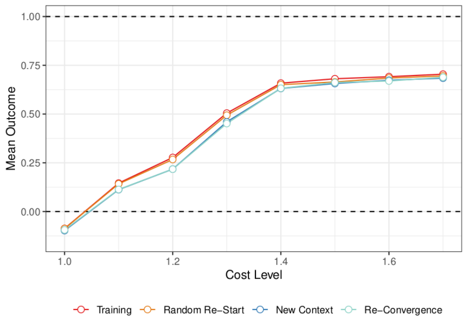
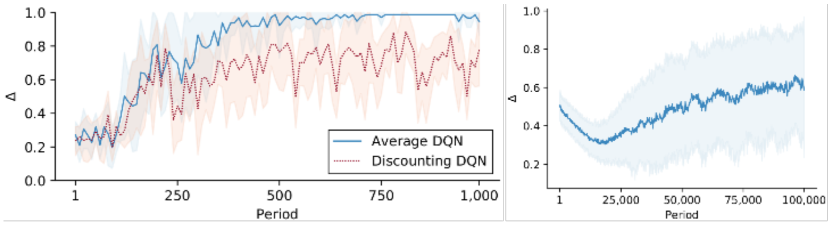


Figure 3: Collusion convergence and re-convergence plots in restricted observation space during training and evaluation in new context. *(Image Credit: Eschenbaum et al.)*

From a game theory interpretation perspective this study also plots transitive closure of the strategy pairs. As after the convergence, theoretically an agent plays pure strategy having a single action corresponding to each observation state which can be discretely represented and plotted. In Figure 2, each node represents an observation state, and the edge represents the transition that occurs from the strategy pair for the agents in the joint policy. We can see different variations in these plots for the same parameterization context if the initial seeds are different but all of them are robust to defections. Further, the strategy pairs are also evaluated in different testing contexts, where we observe overfitted complex transitive closure plots as highlighted in Figure 2.b. With these different transitive closure plots observations, it is concluded that agents are not learning the complete convergence strategies/policies but rather only learn their approximations. Therefore, the authors attempt to reduce the policy strategy space where agents only observe their own past action price. Figure 3, demonstrates average profit gains for different cost prices with this reduced state formulation. We observe that with this state simplification, agent's collusion propensity increases across all the evaluation contexts when marginal cost prices are high. This highlights the increased tendency to gain more rewards when high profit gains are involved otherwise the agent prefers the equilibrium state.

4 Algorithmic Collusion: Insights from Deep Learning

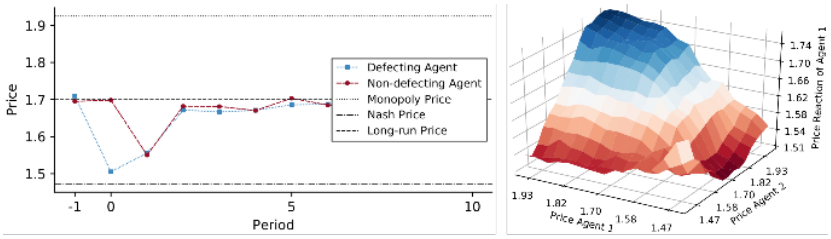


a.) Profit gains for discounted and average reward setup *(avg. 10 runs)*

b.) Average profit gain in DQN duopoly baseline *(avg. 30 runs)*

Figure 4: Average profit gain curves for measuring collusion propensity in baseline duopoly markets. *(Image Credit: Hettich et al.)*

This study conducts similar experiments as the first study to measure different aspects of collusion propensity and robustness. In the previous study, the collusion converged approximately after a few million timestep iterations across different contexts. On an average with DQN model usage in the current study it converges around at 850,000 timesteps. Clearly, with usage of DQN models the convergence happens relatively faster, as more parameters are available for better approximation of policies. Second from Figure 4.a, we also observe that loss function reformulation by using average rewards for averaged DQN learning provides more stabilization in the learning process. Whereas, the discounting DQN keeps on looking for a better partial order policy set and keeps on overfitting. This further destabilizes the learning process which results in oscillating behavior around several lower optima values. The DQN implementation consists of exploration factor ε, which when decreases the agent actions become less explorative, which in turn stabilizes convergence curves. The Figure 4.b, shows that as part of initial exploration strategy the agents often opt for price undercuts leading to almost static Nash equilibrium. And, finally with decreasing ε the agents turn towards collusion convergence in the given duopoly baseline setting.

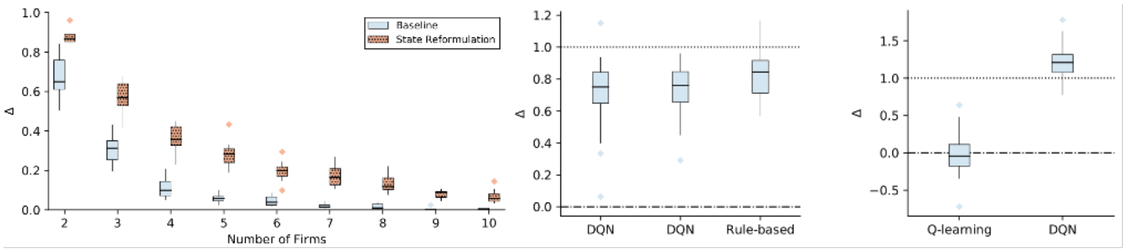


a.) Price reaction after agent defection

b.) Learning strategy in duopoly

Figure 5: Agent strategy analysis and interpretation summary. *(Image Credit: Hettich et al.)*

Supra-competitive prices highlight that either a firm's competitive advantage or anti-competitive market behavior has driven market goods to non-sustainable prices. In this problem's context, it means supra-competitive prices are set when either some agents fail to compete effectively or the agents learn effective reward-punishment policies for collusion. From Figure 5.a, we observe that with added manual Nash equilibrium price defection, the second agent also lowers its price to undo the first defecting agent's advantage. The first defecting agent lower its future prices expecting this punishment, and further both agents iteratively start increasing their prices towards a supra-competitive price setting. The action space size is 15 for each agent in this study, and the average agent strategy for the whole state-space *(15 × 15)* is computed in Figure 5.b. The general symmetry in the surface highlights that DQN agents react similarly to previous prices irrespective of which agent played which price. We can utilize Figure 5.b to interpret Figure 4.b experiment, the first player also defects *(low z-axis price value)* after it has detected punishment state of the second agent *(low x-axis price value)* when collusion with agent one *(high y-axis price value)* already existed. Additionally, the prices also increase when both agents near the Nash price, this further assists the agents to return for higher prices after the action-reaction punishment cycle.



a.) Average profit gains in wide oligopolies

b.) Average profit gains in heterogeneous setup

Figure 6: MARL mobile robot framework pipeline for warehouse management. *(Image Credit: Hettich et al.)*

Previously, we have seen collusion is strongly affected by the number of competitors, and the difference in *'context'* i.e. heterogeneity in agent parameterization. With Figure 6.a, we observe that here also with DQN, collusion is harder to sustain as the number of agents increases. But, the collusion capacity is still larger here in comparison to Q-table learning discussed in the first study. More importantly with state reformulation simplification, which considers only overall price and whether agent has defected results in collusion propensity improvement as highlighted in 6.a. Further Figure 6.b elaborates on the experiment analyzing the impact of heterogeneous agents in automated collusion. First from the right subplot in FIgure 6.b, we observe that simple baseline Q-table learning is unable to compete with a relatively novel DQN approach. Additionally, in the left subplot with another added simple price mimicking rule-based agent, the collusion profit gains are unaffected and the collusion index matches duopoly scenarios even with added extra agent. Like the first study, this experiment

Table 1: Functional description of the different algorithms used for collusion experimentation tasks.

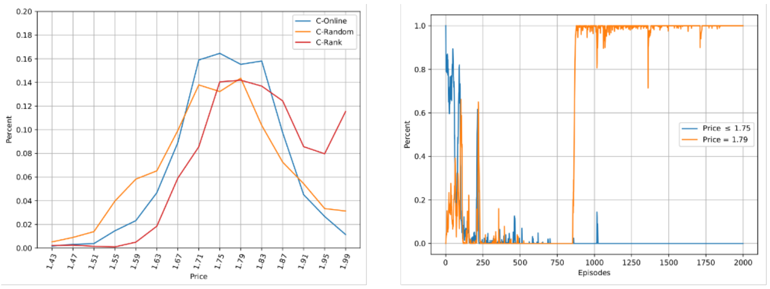
Algorithm Name	Functional Description
C-Random	Randomly updates the previous experience tuple cell
C-Online	Updates the current observed state cell
C-Rank	Updates the state cell when profit does not exceed competitor
D-Random	DQN updating all cells, doesn't learn order of samples
D-Online	DQN using recent most data from replay buffer
D-Rank	DQN using best reward gain experiences from replay buffer for profit gains

highlights the need for homogeneity in agents for symmetric collusion to exist, or otherwise some agents might end up becoming relatively incompetent.

5 Algorithmic Collusion with Experience Replay

In past studies it has been observed that strong temporal correlation from the past helps in maintaining environment stationarity, whereas random experience replay leads to non-stationarity [15, 5]. This study utilizes this insight to deploy a replay buffer that assigns more value to near past experiences that yield higher average profit gains. Like the above discussed studies, this study also compares different aspects like collusion propensity, robustness, and heterogeneity in their experimentation. The Table 1 gives a functional description of the handy algorithm notations used in this study during the different experiments.

From Figure 7.a, we observe that Q-Table based algorithms after convergence do select higher prices in general. The percentage axis highlights the fraction of simulation runs which converge at a given average price point for C-Online, C-Random, and C-Rank algorithms. Similar to findings in the first study, the C-Online equivalent baseline does set high prices for a relatively large number of simulation runs. But, C-Random with random updates and C-Rank with profit gain based updates are able to explore higher price convergence prices. With C-Rank algorithm's capability to find highest prices for relatively larger numbers of simulation runs, its defection behavior in deep learning setup is also analyzed. From Figure 7.b, we observe that D-Rank quickly converges to supra-competitive prices by exploiting the knowledge from experiences where the agent fails to gain profits. Clearly, the fraction of prices below 1.79 drops immediately under 1000 episodes and both players settle for very high prices to get large profit gains. Additionally, the greedy policy matrix values highlighted in the second study are all at high constant value. Which means that after defection, both agents immediately restore their pre-deviation price without incurring any punishment. The non-defecting agent optimistically keeps the commodity price values high to tempt the agent to immediately switch to pre-deviation collusion price.



a.) Long-run prices distribution after convergence

b.) Rank experience replay profit gains variations

Figure 7: Rank experience replay price and profit gain variations summary. *(Image Credit: Han et al.)*

Table 2: Cost asymmetry where marginal cost of efficient Player 2 reduces to 0.5.

Strategy	C-Online	C-Rank	D-Random	D-Rank
Player 1	1.95	1.55	1.40	1.65
Player 2	1.70	1.60	1.25	1.75

Table 3: Convergence result summary under different initializations with larger action space having higher upper bound than monopoly prices.

Initialization	Baseline	Q = 0	Random	Q = 19	Topmost
D-Random	Bertrand	Bertrand	Bertrand	Bertrand	Bertrand
D-Rank	High	High	High	High	High
C-Rank	High	Upper Bound	Upper Bound	High	Volatile

For heterogeneous agent comparison parameterized by different algorithms, it is observed that more novel algorithms are able to increase their corresponding average prices relatively easily. Specifically, the deep variants of the algorithms outperform their Q-table counterpart in terms of their individual profit gains. This already confirms the correlation trend between algorithm novelty and collusion propensity that is observed in the above discussed studies. Additionally, experience ranking algorithms dominate random and online agents to set supra-competitive prices. This form of collusion where one entity clearly drives profits with its algorithmic advantage causes legal ambiguity in a real world setting. Even though collusion persists practically, legally we can quantitatively calculate: i.) which firm to what extent is responsible for colluding, ii.) and what exactly should be the methodology to specifically penalize a firm.

Similar to previous studies as observed from Table 2, the asymmetry in firms reduces the average profit gains when compared with symmetric firms experiments. The inefficiency of the D-Random algorithm still persists as it maintains prices close to Bertrand equilibrium. Whereas, C-Online algorithms like shown in previous discussed studies suffer from collusion profit gain reduction to a limited extent. Whereas, for C-Rand and D-Rank algorithms the more efficient firm prioritizes its profits rather than increasing the average profit gains.

Different initialization yields different optima in deep learning problems, therefore it is important to analyze which initialization is optimal or truly non-informative [20]. The Table 3 highlights different initializations for Q-values, namely: i) baseline He initialization values, ii) zero Q-values, iii) uniform random values, iv) large constant values, v) zero Q-values except for highest price entries. [8]. The deep network based algorithms are robust to all initializations, the D-Random still fails to learn collusion whereas the D-Rank always learns to collude. The C-Rank shows high sensitivity to initialization values, like it learns to collude for baseline initializations similar to above two discussed studies. For the C-Rank algorithm, *'Upper Bound'* convergence represents above monopoly prices, and *'Volatile'* convergence expresses instability in learning prices. The C-Rank shows high dependency on the convergence properties based on priori information. Essentially meaning that the C-Rank algorithm owing to its high temporal correlation issue is vulnerable to settling for local minimas.

6 Discussion and Conclusion

With these studies by using a standard benchmark environment and similar experiments, we can clearly observe the feasibility of collusion in real market systems. [2] Similar experiments testing collusion propensity, robustness, and heterogeneous agent effect across three RL algorithm implementations,

[2] Bibliography papers available at the repository github.com/arana-initiatives/ai-portfolio-bibliography.

helps us in systematically analyzing the collusion systems. First, we observe that with increased novelty the collusion convergence drops drastically from 850k to 2K timesteps. Second, with deep learning based systems the collusion can exist in wider oligopolies as well, and the propensity increases with representation simplification. Further, the experiments highlight that the market agents are capable of learning different strategies for collusion re-convergence. For example, first demonstrating reward-punishment behavior after agent price defection, and second showcasing optimistic high price behavior when ranked replay information is available. Finally, for heterogeneous agents we observe that more novel agents drive the collusion to gain more profit. But, the less novel agents are also forced to place prices above the equilibrium price. These experiments highlight that collusion with simple deep learning based RL algorithms, like DQN can help achieve collusion in minimalist formal economic environments. The practical application translation of such insights still needs to be further analyzed considering different realistic aspects. For example, *'sim-to-real'* or *'concept drift'* challenges might arise as the economic environment does not operate in isolation, but rather shows dependencies to other markets as well [16, 24].

Realistically, there are several limitations for the discussed studies, and the limited insights that these studies provide for the algorithmic collusion task. First, the economic benchmark environment models only Bertrand competition which just varies the prices, but realistically firms can also vary their quantities like in Stackelberg competition [11]. Additionally, the prices selected are from discrete and limited intervals which simplifies the problem significantly which might not be the case in real life scenarios. Second, the actions are executed from a centralized controller, and agents observe the whole system in a given interval. Practically, the agent firms would not have access to the exact reward scheme defined by other competitive firms for tuning their pricing algorithm. This simple setting might be problematic as economic markets being related to other markets might not observe the exact underlying state, and the agents in the multi-agent systems will have their own executor unit. The above listed studies do not discuss the economic systems in such extensive detail algorithmically but rather focus on the interpretability of selected policies. The stability and interpretation insights of these policies, although useful, might not generalize over pricing systems built with deeper and more novel blackbox RL algorithms. The insightful developments from deployment of these advanced systems are not measured in the existing algorithmic collusion literature. Further, the algorithms used in the studies are also not the most novel algorithms being used by any industry or academia standard. Therefore, it might be possible that more novel algorithms can assist agents in even faster collusion, and help firms learn even better agent defection reaction strategies. This would be highly problematic for regulatory authorities, as even after breaking the collusion, the firms would be capable of formulating a new near immediate automated collusion strategy. Finally, only the first study extensively explores the impact on collusion propensity by evaluating already trained agents on different contexts i.e. baseline zero-shot learning performance, and few-shot learning performance for re-convergence. Hence, more experiments studying zero-shot and few-shot collusion performance would give more insights into the robustness of deep learning based RL models for collusion re-convergence.

Practically, it is still incredible that agents are able to collude at such a fast pace without any explicit communication and coordination protocol. Further, to supplement the algorithm performance for ever changing markets, meta-learning can also be explored as an additional information supplementation module for collusion [22]. Since, the product and its associated price life cycles are small for a huge range of commodities. Hence, it would be important to study and test automated collusion algorithms that allow immediate collusion restarts and direct transference across different products and markets. Human colluders and price setters are also involved in the pricing markets, and experiments involving heterogeneous human agents would help to quantify the capabilities of the collusion algorithms more accurately. From a regulation perspective it is very hard to detect such collusions without the knowledge of algorithms, architectures and parameterizations of the models involved. Therefore, the regulatory bodies must enforce regulations on firms to disclose their algorithms to supervising bodies to encourage fair competition [6]. Second, the legal framework for determining liability of firms involved in such malpractices should also be well-defined with minimal ambiguity. Since, in heterogeneous agent environments more novel algorithms drive the collusion but other firms also charge higher prices. Therefore, the extent of firms involved voluntarily or involuntarily should be determined, and based on antitrust behavior and profit gains intent the liability should be determined. With huge amounts of data the ideal collusive strategies, economically can drive the prices to near monopolistic prices with automated pricing algorithms. Therefore, these anti-competent practices should also be regulated under the monopoly antitrust laws as well [17].

With increasing novelty in algorithms across studies, we observed that the firms can collude at a much faster and practically reasonable rate. The current literature certainly analyzes the situation in a simplistic setting with baseline RL models. But, current RL literature and algorithms are capable of handling far more realistic and complex scenarios at a much faster rate. For automated collusion with involvement of heterogeneous parties, determining the extent of responsibility and liability for increased collusion propensity is also ambiguous. Therefore, it is safe to assume that near future deployment of these algorithms in different commodity markets will be a major problem. In conclusion, these automated algorithms do pose serious threats in future to generate collusive market strategies, and promote antitrust behavior amongst the involved firms. Hence, involvement of supervising authorities for regulations, more transparency around usage of these algorithms, and design of democratized platforms that discourage such collusion efforts is very important [18].

References

[1] Assad, S., Clark, R., Ershov, D., and Xu, L. Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market.

[2] Calvano, E., Calzolari, G., Denicolò, V., Harrington Jr, J. E., and Pastorello, S. Protecting consumers from collusive prices due to ai. *Science 370*, 6520 (2020), 1040–1042.

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

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

[5] Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H., Kohli, P., and Whiteson, S. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning* (2017), PMLR, pp. 1146–1155.

[6] Glauner, P. An assessment of the ai regulation proposed by the european commission. In *The Future Circle of Healthcare: AI, 3D Printing, Longevity, Ethics, and Uncertainty Mitigation*. Springer, 2022, pp. 119–127.

[7] Han, B. Understanding algorithmic collusion with experience replay. *arXiv preprint arXiv:2102.09139* (2021).

[8] He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1026–1034.

[9] Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2018), vol. 32.

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

[11] Julien, L. A. A note on stackelberg competition. *Journal of Economics 103* (2011), 171–187.

[12] Kirk, R., Zhang, A., Grefenstette, E., and Rocktäschel, T. A survey of generalisation in deep reinforcement learning. *arXiv preprint arXiv:2111.09794* (2021).

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

[14] Kokkoris, I. A few reflections on the recent caselaw on algorithmic collusion. *Competition Policy International, Antitrust Chronicle, July* (2020).

[15] Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv preprint arXiv:1702.03037* (2017).

[16] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering 31*, 12 (2018), 2346–2363.

[17] Ma, J. *Regulating Data Monopolies.* Springer, 2022.

[18] Noothigattu, R., Bouneffouf, D., Mattei, N., Chandra, R., Madan, P., Varshney, K. R., Campbell, M., Singh, M., and Rossi, F. Teaching ai agents ethical values using reinforcement learning and policy orchestration. *IBM Journal of Research and Development 63*, 4/5 (2019), 2–1.

[19] Papoudakis, G., Christianos, F., Rahman, A., and Albrecht, S. V. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737* (2019).

[20] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (2013), PMLR, pp. 1139–1147.

[21] Sutton, R. S., and Barto, A. G. *Reinforcement learning: An introduction.* MIT press, 2018.

[22] Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., and Botvinick, M. Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience 21*, 6 (2018), 860–868.

[23] Zhang, S., and Sutton, R. S. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275* (2017).

[24] Zhao, W., Queralta, J. P., and Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)* (2020), IEEE, pp. 737–744.