On Mechanism Underlying Algorithmic Collusion*

Zhang Xu† Wei Zhao‡

December 5, 2024

Abstract

Two issues of algorithmic collusion are addressed in this paper. First, we show that in a general class of symmetric games, including Prisoner's Dilemma, Bertrand competition, and any (nonlinear) mixture of first and second price auction, only (strict) Nash Equilibrium (NE) is stochastically stable. Therefore, the tacit collusion is driven by failure to learn NE due to insufficient learning, instead of learning some strategies to sustain collusive outcomes. Second, we study how algorithms adapt to collusion in real simulations with insufficient learning. Extensive explorations in early stages and discount factors inflates the Q-value, which interrupts the sequential and alternative price undercut and leads to bilateral rebound. The process is iterated, making the price curves like Edgeworth cycles. When both exploration rate and Q-value decrease, algorithms may bilaterally rebound to relatively high common price level by coincidence, and then get stuck. Finally, we accommodate our reasoning to simulation outcomes in the literature, including optimistic initialization, market design and algorithm design.

Keywords: Algorithmic Collusion; Artificial Intelligence; Q-learning; Stochastic Stability; Edgeworth Cycle; Market Design.

*We acknowledge Tristan Tomala, Nicolas Vieille, Marco Scarsini, Arthur Dolgopolov, Jean-Edouard Colliard, Satoru Takahashi, Michihiro Kandori, Yu Zhu and Zhiwei Cui for their helpful comments and discussions. We also acknowledge seminar and conference participants at GAIMSS, Stony Brook Game Theory Conference, Shandong University, GAMES 2024 and China Academy of Science for helpful comments. All errors are our own.

†School of Economics, Renmin University of China, China. *Email*: xuzhang@ruc.edu.cn

‡School of Economics, Renmin University of China, China. *Email*: wei.zhao@outlook.fr

1 Introduction

The core of market in its efficiency in resource allocation relies on competition among firms. Firms' collusive behaviors increase prices and therefore shut down potential trades which benefit both consumers and firms. Limiting firms' collusion in real operations relies on detecting communications among managers in secret chambers, since cost and demand functions are unknown to regulators. Nowadays, firms growingly rely on algorithms to make pricing decisions. It has been well documented that learning algorithms may adapt to tacit collusion in simulations, which therefore poses challenges to the existing policy-makers. To regulate artificial intelligence's collusive behavior, the first step is to understand the mechanism underlying algorithmic collusion.

In this paper, we try to address two issues related to (learning) algorithmic collusion. First, whether algorithmic collusion is driven by (1) adapting to some mechanisms/strategies maintaining collusive outcomes; (2) or just failing to learn Nash Equilibrium (NE) due to insufficient explorations. Economists' opinions on this issue diverge. Second, though the "stability" of final convergent outcome has been extensively studied in the literature, it has been rarely discussed on the learning process illuminating how algorithms adapt to collusive outcomes, in real simulations with insufficient learning.

To address the first issue, we construct and prove the uniqueness of the stochastically stable state with (strict) NE on the path in a class of symmetric games. In these symmetric games, two players share utility function with the same ranked action space. Given both players choosing same actions, their utility strictly increases in the rank of chosen actions. On one hand, if one player deviates to some strictly lower actions, then the other player strictly hurts and derive the minimal utility. On the other hand, there always exists a strictly lower action reaching weakly higher utility, as long as their mutually chosen action is not the lowest. Finally, their utility increases if their opponents increases their actions. Note that Bertrand Competition, Prisoner's Dilemma and any nonlinear mixture of first and second price auction (without information asymmetry) fall into this class of games. Besides, bilaterally choosing the lowest action is the unique strict NE though multiple NE may exist.

The learning agent maintains a vector of Q-values that encode her subjective discounted payoffs from taking the corresponding actions. She typically chooses the action with the highest Q-value but occasionally experiment with other actions according to a predetermined rule. Upon receiving a new discounted payoff, she updates the Q-value associated with the chosen action, making it closer to the new discounted payoff. This class of learning programs includes Q-learning.

In the first step, we excludes the possibility of occasional experimentation and the transition of the pair of Q-values therefore forms a Markov process without perturbation. We show that all the recurrent states are absorbing, i.e. all recurrent classes are singletons. Besides, any pair of symmetric action profile can be chosen forever under some absorbing states. On one hand, this result echoes with Asker et al. (2023) that collusive outcome may be reached in the long run whenever the exploration rate vanishes. [1] On the other hand, this result also explains that cycles do not emerge in real simulations.

In the second step, we consider the scenario with vanishing possibility of experimentation. Fixing the probability of experimentation, the transition forms a Markov chain with a unique recurrent class and therefore a unique stationary distribution. A state is stochastically stable (introduced by Foster and Young 1990) if it is in the support of the limiting stationary distribution as the experimentation probability vanishes. Any stochastically unstable state will not be sustained in the long-run subject to persistent and slight enough perturbations/explorations. In this regard, we refer stochastically stable states as outcomes under sufficient learning. We prove that the absorbing state s^N, where both play the lowest action and which is immune to unilateral deviations, is the unique stochastically stable state. For one thing, this result shows that collusive outcomes observed in simulations will no longer appear in the long run given additional and persistent explorations. In other words, they are driven by failure to learn NE due to insufficient learning. For another, this result complements to the literature on multi-agent reinforcement learning, where results on its convergence are absent. Our result shows that, within the class of symmetric games, the repeated interactions between Q-learning algorithms converges to the strict NE with diminishing fixed exploration rate. The proof of the result relies on the equivalence between stochastically stable state and minimum spanning tree, developed by Kandori et al. (1993) and Young (1993). The core of the proof is to construct a strict partial order on set of absorbing states such that (1). the absorbing state s^N is the lower bound of the set; (2). for any other absorbing states, there exists a unilateral one-step mutation such that the state will be transited to another absorbing state ranked below. In other words, start from any other absorbing state, we can always find a succession of one-step mutations along which the absorbing state s^N will finally be reached, which excludes the possibility that the process will be trapped in cycles of absorbing states. Comparing to games with only two actions, the strict partial order, for the class of symmetric games with multiple actions, turns out to be not that intuitive since some state may transit to another state with strictly higher selected

[1] In Asker et al. (2023), they show that all prices can be achieved, but their proof relies on an unreasonable initialization of Q-values, whereby Q-values are set below the minimum discounted utility of the corresponding action, which is excluded in our analysis.

actions. We will come back to the illustration of the order when addressing the second issue.

As suggested by our theory, the collusive outcome observed in real simulations is led by failure to learn NE due to insufficient learning in real simulations. Then, how the reinforcement learning algorithms adapt to collusive outcomes during the learning process? To address the issue, we first replicate past simulations in a Bertrand competition environment with exponentially decay exploration rate. Consistent with the past literature, relatively high collusive degree is observed for a wide range of learning rate and decay rate, fixing a relatively high discount factor. However, up to our knowledge, we are the first to point out that the convergent prices are crucially dependent on the discount factors, even if memory of past histories is deactivated. The collusive level decreases as the discount factor drops. Whenever the discount factor decreases to zero, so does the collusive level. Besides, simulations with constant but minute exploration rate are also conducted to test our theory. Each session of experiments are run for a fixed number of episodes. The convergent outcome is determined by the average price in the last 10000 episodes, weighted by frequency of occurrences. We find that, when discount factor is below a relatively high threshold 0.75, Q-learning agent will adapt to the strict equilibrium profile, consistent with our theory.

We then propose mechanisms underlying algorithmic collusion, to accommodate the outcomes in real simulations. As is suggested by common wisdom, reinforcement learning algorithms adapt to NE based on sequential and alternating price undercut. However, this price undercut process will be constantly interrupted since Q-values of low prices are inflated *for some reason*. Hence, even if winning agent grab all the market share by undercutting price, the generated profit is too low to reinforce its choice of undercut price in the future. Then, both the winning and losing agent have to bilaterally rebound to some higher prices. The price undercut process will then be iterated, making the price curve like an edge-worth cycle. When these Q-learning agents bilaterally rebound to the same relatively high price level coincidentally, the Q-value of this high price increases while the Q-values of the rest prices stay the same. Therefore, a strictly positive gap emerges and can sustain unilateral deviations within a given periods, as the exploration rate decreases to a low enough scale. Finally, we argue that the discount factor plays a critical role in the inflation of Q-values. Note that Q-values are initialized in a plausible range such that each price are reinforced as long as it wins the market game. However, when Q-learning agents update their Q-values, the strictly positive discount factor always injects the maximal Q-value, serving as the estimation for the discounted future payoff, and the bubble of Q-values therefore emerges. In early episodes, actions are mainly chosen by exploration and then discount factor inflates Q-values when Q-learning agents update Q-values of choosing actions. One way to solve

the inflation problem is to impose a lower bound on the prices each firm can quote. When quoting the minimal price and winning the market game, the generated profit is still high enough to reinforce its choice of the minimal price in the future, facing the inflated Q-values. One policy recommendation for regulators is that allowing for low algorithmic collusion can exclude high collusion.

We finally accommodate our mechanism to simulation outcomes in existing literature. Asker et al. (2022) and Asker et al. (2023) replaces the exogenous exploration by initializing Q-values at relatively high level, named as "optimistic initialization". The collusive outcome observed in the simulation is then driven by artificial inflation of Q-values during initialization. Banchio and Skrzypacz (2022) draws a comparison between First and Second Price Auction and finds that Q-learning agents adapt to NE among all sessions in the Second Price Auction (SPA). Note that in SPA, the profit for the winner only depends on the bid quoted by the loser. Therefore, a unilateral rebound, instead of bilateral rebound, will follow the interruption of price undercut process. Both learning agents will be trapped in alternatively choosing the highest bid, until both choosing the highest bid can be stable. Asker et al. (2022) also proposes that synchronous updating can completely eliminate algorithmic collusion and partially synchronous updating can alleviate collusion. The synchronous updating, through constantly updating Q-values for low prices even if not chosen, can squeeze the bubble of their Q-values and therefore the price undercut process will be continued without interruption.

1.1 Literature Review

The theoretical part of our paper contributes to the literature on stochastic stability, a concept introduced and developed by Foster and Young (1990), Kandori et al. (1993), and Young (1993). These foundational works use the concept of "stochastically stable states" or "long-run equilibria" to select equilibrium in games with multiple equilibria. Over time, various papers have explored questions related to the speed of convergence and the properties of intermediate outcome (e.g., Ellison 2000; Cui and Zhai 2010; Levine and Modica 2016). Additionally, this theory has been applied to the study of evolutionary dynamics in network games (e.g., Jackson and Watts 2002; Staudigl and Weidenholzer 2014). In our paper, we apply the concept of stochastic stability to select absorbing states (which are not necessarily equilibria) in Markov processes induced by reinforcement learning. Dolgopolov (2024) was the first to formally use stochastic stability to analyze the convergence problem in algorithmic collusion, specifically within the context of the Prisoner's Dilemma. He demonstrated that

Q-learning with ε-greedy exploration leads to convergence to a competitive outcome. We extend this result to a more general class of games and a broader set of learning algorithms (with discount factor).

The simulation part of the paper contributes to the recently emerging literature on algorithmic collusion through simulations. Based on whether memory of one-period price history is activated or not, we divide this literature into two strands. In the first strand, the seminal paper by Calvano et al. (2020) show that the Q-learning agents, taking the last period price history as the state variable, seems to adapt to grim trigger strategy to sustain high collusive level, under ϵ-greedy exploration mode. Various settings have been examined and similar results have been derived, including sequential pricing (Klein 2021), algorithmic trading in financial market (Dou et al. 2023), platform governance (Johnson et al. 2023), imperfect monitoring (Calvano et al. 2021), trust game (Wu et al. 2023), interaction between algorithm and human (Werner 2023). The second strand, where the memory is deactivated, is initiated by Waltman and Kaymak (2008), who show that Q-learning agents with Boltzmann exploration mode may end up with collusion in Cournot competition even if memory is deactivated.[2] The following literature mainly focuses on ϵ-greedy exploration mode but still derive collusive outcome in various settings, including dynamic storage optimization (Abada and Lambin 2023), competition between market maker in financial market (Colliard et al. 2022), dynamic storage management (Abada and Lambin 2023), comparison between First and Second Price Auction (Banchio and Skrzypacz 2022), price discrimination (Xu et al. 2024), optimistic initialization and algorithm design (Asker et al. 2022, 2023). Our paper belongs to the second strand. Comparing to the existing literature, which just document collusive outcomes in specific settings, our paper tries to explain how reinforcement learning algorithms adapt to collusion and accommodate our reasonings to various simulation outcomes in the literature. Besides, though the "stability" of the final convergent outcome has been extensively studied, [3] our paper, on the other hand, focuses on the learning process and illustrates what components and factors during the process drives the profit-maximizing learning algorithms to coordinate with each other.

[2] In his paper, he also consider the scenario with memory and show similar results hold. Even if the discount factor is fixed at zero when memory is not allowed, the collusive outcome is driven by Boltzmann exploration mode, which has been shown analytically by Waltman and Kaymak (2007, 2008) and Dolgopolov (2024).

[3] For instance, Asker et al. (2023) adopts the solution concept of "experience based equilibrium", proposed by Fershtman and Pakes (2012), to explain stability of any symmetric action profile whenever exploration rate vanishes. Banchio and Mantegazza (2023) uses deterministic ODE to approximate the stochastic discrete learning process and shows that "mutual collusion" in Prisoner's Dilemma may be stable under certain conditions.

Perhaps the most closely related paper is Banchio and Mantegazza (2023), who use stochastic approximation technique to approximate the discrete, stochastic process in Prisoner's Dilemma with a continuous, symmetric and deterministic ODE when $\alpha \to 0$. Spontaneous coupling, the key component for mutual cooperation based on their theory, relies crucially on symmetry assumption. On one hand, symmetry induces each agent to estimate the value of cooperation given its opponent's cooperation, and the value of defection given its opponent's defection. On the other hand, after sequences of mutual defection led by unilateral defection, symmetry also makes it possible for simultaneously reverting back to cooperation, crucial for the stability of mutual cooperation. Our paper, on the other hand, proposes mechanisms to accommodate algorithmic collusion in real simulations with $\alpha > 0$, to incorporate stochasticity, discreteness and nonsymmetry into consideration. Our reasoning can also explain the effect of discount factors in real simulations, which is muted in their theory.

2 Preliminaries

Consider a two-player $K \times K$ symmetric game. Each player $i \in I = \{1, 2\}$ has a set of actions $A = \{a_1, a_2, \ldots, a_K\}$. Denote the payoff of playing action $a \in A$ when the opponent plays action $a' \in A$ as $u(a, a')$.

Assumption 1. *Assumptions about $u(\cdot, \cdot)$:*

1. *on-diagonal:* $\underline{u} < u(a_1, a_1) < u(a_2, a_2) < \ldots < u(a_K, a_K)$;
2. *off-diagonal:* $u(a_k, a_{k'}) = \underline{u}$, *if* $k' < k$;
3. $\forall k > 1$, $\exists k' < k$, *s.t.* $u(a_{k'}, a_k) \geq u(a_k, a_k)$;
4. $\forall k$, $u(a_k, a_{k'}) \leq u(a_k, a_{k''})$ *if* $k'' > k'$.

In this class of games, there may be many equilibria, but (a_1, a_1) is the unique strict NE. For convenience, we say $a > a'$ if the index of a is higher than that of a'. Assumption 1.1 defines an order of actions, where the lowest action is the "Nash action", inducing the lowest utility among all symmetric action profiles. If any learning algorithm realizes expected payoffs strictly higher than that under strict NE, then we call this as algorithmic collusion. Assumption 1.2 states that when a player chooses a higher action than their opponent, she will lose the game and can only obtain the value of the outside option. Assumption 1.3

states that whenever players cooperate to achieve a high value, there exists a lower action, and deviating to this lower action is weakly better off. Assumption 1.4 shows that given their own action, when the opponent raises her action, the player will not be worse off.

Example 1. *Prisoner's Dilemma. In the context of the Prisoner's Dilemma, let D represent defect and C represent cooperation. The condition $u(C,D) < u(D,D) < u(C,C) < u(D,C)$ satisfies all the assumptions above.*

Example 2. *Generalized Combination of First and Second Price Auction. When a bidder bids b_1 and wins (the highest bidder wins, and if bids are tied, each bidder gets half of the item), assuming the second highest bid is b_2, the winning bidder only needs to pay $p(b_1, b_2)$. Here, $b_2 \leq p(b_1, b_2) \leq b_1$ and it satisfies $\frac{\partial p}{\partial b_2} \geq 0$.*

In the discrete case of this auction, if the highest bid is allowed to be slightly lower than the value of the item v, i.e., $A = \left\{0, \frac{v}{K}, \ldots, \frac{(K-1)v}{K}\right\}$, then the highest bid $\frac{(K-1)v}{K}$ becomes the unique strict NE. Here, $a_1 = \frac{(K-1)v}{K}$, $a_2 = \frac{(K-2)v}{K}$, and so on.

It is clear that Assumptions 1.1 and 1.2 are satisfied. When $K \geq 2$, $p\left(\frac{kv}{K}, \frac{kv}{K}\right) = \frac{kv}{K}$ and $p\left(\frac{(k+1)v}{K}, \frac{kv}{K}\right) \leq \frac{(k+1)v}{K}$. Therefore, $u\left(\frac{kv}{K}, \frac{kv}{K}\right) = \frac{1}{2}\left(v - \frac{kv}{K}\right) \leq v - \frac{(k+1)v}{K} \leq u\left(\frac{(k+1)v}{K}, \frac{kv}{K}\right)$, which satisfies Assumption 1.3. When player 1 wins and player 2 lowers her bid, due to $\frac{\partial p}{\partial b_2} \geq 0$, player 1's utility is weakly increasing. Hence, Assumption 1.4 is satisfied.

Example 3. *Bertrand Competition. In the discrete case of Bertrand competition, when the lowest price is set slightly higher than the marginal cost, this lowest price becomes the unique strict NE. Bertrand competition is equivalent to the first price auction. Therefore, it is a special case of the above example of an auction.*

Two model-free reinforcement learning algorithms are delegated to repeated play a fixed game in this class. Without any reference to the underlying environment, each algorithm repeatedly interact with the environment and update estimated payoffs based on observed generated payoffs, which then improve the accurateness of their decision making. At a given t, denote Q_i^a the learning agent i's estimated discounted payoffs if choosing action a at the current period, and $\mathbf{Q}_i = (Q_i^a)_{a \in A}$ the associated vectors. If both the decision and updating rule depends only on the current Q-values, then the underlying stochastic process is a Markov chain where state s is the pair of current Q-vectors $s = (\mathbf{Q}_1, \mathbf{Q}_2)$. Denote the set of all possible states as $\mathcal{S}$. To take into account the machine precision limitations, it is assumed that all Q-values lie in a fine grid with some constant $\eta > 0$ spacing. Formally, $\mathcal{S} \subseteq \{(\mathbf{Q}_1, \mathbf{Q}_2) : \mathbf{Q}_i \in \mathcal{D}^{|A|}\}$, where $\mathcal{D}$ represents the compact set $\{\zeta\eta, \zeta \in \mathbb{Z}\}$. Assume η is small enough and $u(a, a')/(1-\delta) \in \mathcal{D}$ for any $a, a' \in A$. Whenever a Q-value does not align

with this grid, it is rounded to the nearest grid point. Therefore, the set $\mathcal{S}$ is always finite with fixed cardinality, for any fixed level of precision η.

Throughout the paper, we impose the assumption that Q-values is weakly higher than the lowest attainable discounted payoffs, i.e. $\mathcal{S} = \{(\mathbf{Q}_1, \mathbf{Q}_2) : \mathbf{Q}_i \in [\underline{u}/(1-\delta), \bar{Q}]^{|A|} \cap \mathcal{D}^{|A|}, \forall i\}$, for any $\bar{Q} \geq \max_{\mathbf{a}} u_i(\mathbf{a})/(1-\delta)$.[4] The main reason is that, under a greedy policy with some exploration (e.g., ε-greedy policy, Boltzmann exploration, etc.), any states outside the interval will enter the interval given finite number of explorations, while states in the interval will never leave the interval. Hence, the recurrent classes will always be inside the interval. It is without loss to restrict to this interval when characterizing the stochastically stable state. [5]

3 Stochastic Stability

In this paper, we mainly focus on Q-learning algorithms with ε-greedy exploration mode. One can refer to Appendix A for a detailed introduction. Note that it has been well studied that Q-learning algorithms with Boltzmann exploration mode (see footnote 22) may converge to collusion even in Prisoner's Dilemma (c.f. Waltman and Kaymak 2008, Dolgopolov 2024). Besides, most simulation analysis in recent literature adopt the ε-greedy exploration mode. The main result of this section is that the long-run probability of playing any action profile other than the strict NE will vanish as the exploration converges to zero. Therefore, we conclude that algorithmic collusion is due to insufficient learning in real simulations. One can skip this section if they are more interested in how learning algorithms adapt to collusive outcomes in real simulations.

3.1 Unperturbed Dynamics

If player i plays a^i at period t when her opponent plays a^{-i}, then her *estimated* discounted payoff of a^i is

$$V^{(a^i, a^{-i})}(s_t) = u(a^i, a^{-i}) + \delta \max_{a' \in A} Q_i^{a'}(s_t), \tag{1}$$

[4]The upper-bound $\bar{Q}$ is imposed to ensure the finiteness of set of Q-values. The main reason that we do not impose other restrictions on the upper bound is to allow for optimistic initialization conducted by Asker et al. (2022) and Colliard et al. (2022).

[5]For all states outside the interval, we can construct minimal paths that connect to the states in the interval. In constructing the minimum spanning tree (see Subsection 3.2), we can completely ignore states outside the interval since they are just attached branches of the first connected state in the interval, which is fixed no matter how states in the interval are connected.

where $\delta \in [0,1]$ is the discount factor. On the right-hand of Equation (1), the first item is her instantaneous payoff and the second item is her estimated payoff she can get in the future under greedy policy, i.e. selecting actions maximizing corresponding Q-values.[6] After playing a^i, player i updates her Q-matrix according to the following rule (when her opponent plays a^{-i}):

$$\begin{aligned} Q_i^{a^i}(s_{t+1}) &= F_i^{(a^i,a^{-i})}(s_t), \\ Q_i^{a'}(s_{t+1}) &= Q_i^{a'}(s_t), \quad \text{for all } a' \neq a^i, \end{aligned} \tag{2}$$

where

$$|F_i^{(a^i,a^{-i})}(s_t) - V^{(a^i,a^{-i})}(s_t)| \leq |Q_i^{a^i}(s_t) - V^{(a^i,a^{-i})}(s_t)| \tag{3}$$

with *strict* inequality if $Q_i^{a^i}(s_t) \neq V^{(a^i,a^{-i})}(s_t)$. Given the current state s_t and the action profile (a^i, a^{-i}), denote the updated state $s_{t+1} := \mathcal{F}^{(a^i,a^{-i})}(s_t)$. In stark contrast with learning with synchronous updating, including best-response dynamics and no-regret learning, reinforcement learning algorithms adopts asynchronous updating, i.e. only updating the Q-values for actions they take. Standard Q-learning algorithms update Q-values as a weighted average of previous Q-values and the estimated discounted payoffs. The updating rule adopted in our paper is a generalization of standard Q-learning algorithms, which only requires the updated value be closer to the estimated discounted payoffs.

We first consider greedy policy as the decision rule, which assigns strictly positive probability on actions with highest Q-values.[7] Denote $\mathcal{A}_i(s) = \arg\max_{a \in A} Q_i^a(s)$ the set of selected actions for agent i at state s for player i, and $\mathcal{A}(s) = \mathcal{A}_i(s) \times \mathcal{A}_{-i}(s)$ the set of selected action profiles. The updating rule together with the greedy policy generate a set $\mathbb{P}_0$ of unperturbed dynamic process as

$$\mathbb{P}_0 = \{P_0 : P_0(s,s') > 0 \text{ if and only if } s' = \mathcal{F}^{(a^i,a^{-i})}(s) \text{ with } (a^i,a^{-i}) \in \mathcal{A}(s)\}. \tag{4}$$

In the context of the unperturbed dynamics, we are particularly interested in the set of recurrent classes. A recurrent class is a set of states where each state within the class is accessible from any other state in the same class, but states from different recurrent classes cannot reach one another. The set of all recurrent classes is denoted by $\mathcal{R}$. A state that belongs to any recurrent class is called a recurrent state. If a recurrent class contains only a single state, that state is known as an absorbing state, as it cannot transition to any other

[6]The Q-learning algorithms with ε-greedy exploration mode is off-policy since its decision rule adopts ε-greedy policy while it estimates its value under greedy policy.

[7]The greedy policy we adopted is a generalization of the standard one, which assigns equal probability on actions maximizing Q-values.

state once reached. To simplify notation, denote $s \in \mathcal{R}$ if s is an absorbing state. For any subset $S \subseteq \mathcal{S}$, denote $\mathcal{A}_i(S) = \bigcup_{s \in S} \mathcal{A}_i(s)$ the set of all possible selected actions for agent i at some state $s \in S$, and $\mathcal{A}(S) = \bigcup_{s \in S} \mathcal{A}(s)$.[8]

Lemma 1. *Fix any recurrent class S, then there exists $(\underline{q}^i, \underline{q}^{-i})$ such that,*

1. *for any $s \in S$, if $|\mathcal{A}_i(s)| = 1$, then for all $a \in \mathcal{A}_i(S) \setminus \mathcal{A}_i(s)$, $Q_i^a(s) = \underline{q}^i$; if $|\mathcal{A}_i(s)| > 1$, then for all $a \in \mathcal{A}_i(S)$, $Q_i^a(s) = \underline{q}^i$;*
2. *for all $(a^i, a^{-i}) \in \mathcal{A}(S)$, at least one of the following equations holds: $\underline{q}^i = u(a^i, a^{-i})/(1-\delta)$, $\underline{q}^{-i} = u(a^{-i}, a^i)/(1-\delta)$.*

[Insert Fig 1]

Lemma 1 means that any recurrent class is attached with a profile of uniform lower bound of Q-values. If an action may not be selected with strictly positive probability at some state in the recurrent class, the corresponding Q-value is supposed to achieve the uniform lower bound. Specifically, as shown in Figure 1, suppose agent i starts to abandon action a^i at state s, then the Q-value $Q_i^{a^i}(s)$ is supposed to exactly hit the lower bound. If an action profile is selected with strictly positive probability in the recurrent class, the uniform lower bound for at least one agent is exactly his discounted payoffs. Note that Lemma 1 holds for any game form, not necessary in the class of games we proposed.

Proposition 1. *For any game in the class, all recurrent states are absorbing.*[9] *Besides, a state s is absorbing in the unperturbed process, i.e., $s \in \mathcal{R}$, if and only if there is an $a \in A$ such that:*

$$Q_i^a(s) = \frac{u(a,a)}{1-\delta}, \tag{5}$$

and

$$Q_i^a(s) > Q_i^{a'}(s) \text{ for all } a' \in A \setminus \{a\}, i \in I. \tag{6}$$

Figure 2 illustrates the key part for the proof of Proposition 1. Suppose a non-symmetric action profile (a^i, a^{-i}) is played with strictly positive probability in some recurrent class S, then the loser will achieve the lowest utility and weakly prefer to choose any other actions. Then $(a_1, a^{-i}) \in \mathcal{A}(S)$ and $(a_1, a_1) \in \mathcal{A}(S)$ for similar reason. This then contradicts Lemma 1

[8] Note that $\mathcal{A}(S) \neq \mathcal{A}_i(S) \times \mathcal{A}_{-i}(S)$.

[9] In Dolgopolov (2024), the statement in Proposition 1, "If the game can be solved by iterated elimination of strictly dominated strategies (IESDS), then such states are the only recurrent classes," is incorrect. Solvability by IESDS does not preclude the existence of cycles.

as no agent's discounted utility of profile (a_1, a_1) can reach the uniform lower bound attached to the class S.

[Insert Figure 2]

A key implication of Proposition 1 is that, for any initial Q-value, the unperturbed dynamics will always end up constantly maintaining some symmetric action profile. Therefore, the pair of Q-learning algorithms may stuck at any symmetric action profile whenever the exploration rate vanishes, which echoes with Asker et al. (2023). However, Proposition 1 further excludes the possibilities of cycles of action profiles in the long run.

In the next subsection, we introduce the possibility of exploration and characterize which absorbing states vanish subject to sufficient but minute exploration rate.

3.2 Perturbed Dynamics

Reinforcement learning algorithms need to explore to learn its interacting environment. The recent literature of simulation analysis often adopts the ε-greedy policy, which chooses the greedy policy with probability $1-\varepsilon$ and a random action otherwise. The perturbed dynamics are induced by this ε-greedy policy.

Assumption 2. *For any $\varepsilon \in (0,1)$, the set of perturbed dynamics is $\mathbb{P}_\varepsilon := \{P_\varepsilon : \lim_{\varepsilon\to 0} P_\varepsilon(s,s') = P_0(s,s'),\ P_\varepsilon(s,s') \in O(\varepsilon^{c(s,s')})\}$, where $c(s,s')$ satisfies:*

1. *$c(s,s') = 0$ if there exists $\mathbf{a} \in \mathcal{A}(s)$ such that $s' = \mathcal{F}^{\mathbf{a}}(s)$;*
2. *$c(s,s') = 1$ else if there exists $(a^i, b^{-i}) \in \mathcal{A}_i(s) \times \mathcal{A}^C_{-i}(s)$ for some $i \in I$ such that $s' = \mathcal{F}^{(a^i,b^{-i})}(s)$;*
3. *$c(s,s') = 2$ else if there exists $(b^i, b^{-i}) \in \mathcal{A}^C_i \times \mathcal{A}^C_{-i}$ such that $s' = \mathcal{F}^{(b^i,b^{-i})}(s)$;*
4. *$c(s,s') = \infty$ if $s' \neq \mathcal{F}^{\mathbf{b}}(s)$ for any $\mathbf{b} \in A_i \times A_{-i}$.*

Several remarks deserves to be drawn on the class $\mathbb{P}_\varepsilon$ of perturbed dynamics. First, the perturbed dynamics induced by the ε-greedy policy satisfy Assumption 2. Second, the perturbations are indirectly imposed on actions rather than directly on states. Third, $c(s,s') \in \{0,1,2,\infty\}$ measures the minimal number of mutations required to transit from state s to state s' in one step. The literature uses the terminology, *the one-step cost of*

moving from s to s', for $c(s, s')$. An implicit assumption is that the probability of perturbing action profile $\mathbf{a}$ to $\mathbf{a}'$ and therefore the transition probability P_ε is independent of the current state s, a defining feature of the ε-greedy policy. In contrast, under Boltzmann exploration, the probability of perturbing an action with high Q-value to the one with low Q-value is relatively lower. This implies that transition from collusive outcome to competition is less likely than that from competition to collusive outcome. Such a "discriminatory" exploration rule introduces an inherent mechanism to converge to a collusive outcome. In this paper, we exclude this channel from our analysis. The following lemma characterizes properties of stationary distribution for these unperturbed dynamics. [10]

Lemma 2. *For any series of perturbed dynamics $\{P_\varepsilon \in \mathbb{P}_\varepsilon : \varepsilon \in (0,1)\}$, we have*

1. *P_ε always admits a unique stationary distribution μ_ε, for any $\varepsilon \in (0,1)$;*
2. *The limit $\lim_{\varepsilon \to 0} \mu_\varepsilon$ always exists and is common for any series $\{P_\varepsilon \in \mathbb{P}_\varepsilon : \varepsilon \in (0,1)\}$.*

Following the solution concept proposed by Kandori et al. (1993); Young (1993), we say that a state s is *stochastically stable* if $\lim_{\varepsilon \to 0} \mu_\varepsilon(s) > 0$. In the long run, the probability of visiting any stochastically non-stable state will vanish as the exploration rate converges to zero.

Denote s^N a specific state where

$$Q_i^a(s^N) = \frac{(1-\delta)u(a, a_1) + \delta u(a_1, a_1)}{1-\delta}, \forall a \in A, \ \forall i \in I. \tag{7}$$

By construction, for any $a \neq a_1$,

$$Q_i^a(s^N) = \frac{(1-\delta)\underline{u} + \delta u(a_1, a_1)}{1-\delta} < \frac{u(a_1, a_1)}{1-\delta} = Q_i^{a_1}(s^N). \tag{8}$$

Therefore, the state s^N has two key properties. First, at state s^N, both agents play a_1 with probability 1 (under greedy policy), i.e. $\{(a_1, a_1)\} \in \mathcal{A}(s^N)$. Second, any unilateral deviation from a_1 does not alter the Q-value of any $a \in A \setminus \{a_1\}$ nor does it affect the action selected by the greedy policy, i.e., the set $\mathcal{A}$ in the subsequent states remains unchanged (Lemma 3). The following theorem, the main result of the section, explicitly characterizes the stochastically stable states.

[10] In previous literature of stochastic stability, the perturbation is directly imposed on states. The perturbed dynamics satisfies irreducibility naturally and therefore always admits a unique stationary distribution.

Theorem 1. *s^N is the unique stochastically stable state.*

Theorem 1 indicates that, in an ideal environment with sufficient but slight exploration, reinforcement learning algorithms are supposed to adapt to the strict NE (a_1, a_1) in the long-run. Hence, the collusive outcomes observed in real simulations are a result of insufficient learning. Given additionally slight exploration, the probability of collusive outcomes in the long run diminishes and vanishes as the exploration rate converges to zero.

3.2.1 Sketch of the Proof

Kandori et al. (1993) and Young (1993) have introduced spanning tree approach to characterize stochastically stable states. First, define the minimal cost of transiting from s to s' with possibly multiple steps as

$$C(s, s') = \min_{(s_0=s, s_1, \ldots, s_L=s') \in X(s,s')} \sum_{l=0}^{L-1} c(s_l, s_{l+1}), \tag{9}$$

where $X(s, s')$ denotes set of all paths between states s and s'. Second, a spanning tree is a directed graph with all recurrent classes (i.e. absorbing states in our games) in unperturbed dynamics as nodes and a single node as root, to which the rest nodes have a single path to connect. The cost of each edge between absorbing state s and s' is $C(s, s')$ and the cost of the spanning tree is the sum of all edges. Denote h_s some s-rooted spanning tree, and $E(\cdot)$ the set of all edges. The total cost of h_s is $C(h_s) := \sum_{(s',s'') \in E(h_s)} C(s', s'')$. $\underline{h}_s$ is the s-rooted spanning tree with lowest cost. Kandori et al. (1993) and Young (1993) have shown that an absorbing state $\hat{s} \in \mathcal{R}$ is stochastically stable if and only if the cost of $\hat{s}$-rooted spanning tree is lowest, i.e. $C(\underline{h}_{\hat{s}}) \leq C(\underline{h}_s)$ for any $s \in \mathcal{R}$.

We then proceed to prove Theorem 1 through three steps:

Step 1. At least two mutations are required to evade from the absorbing state s^N. We formulate the argument through the Lemma 3.

Lemma 3. *After a single perturbation, s^N will eventually return to s^N with probability 1, i.e., $C(s^N, s) \geq 2$ for all $s \in \mathcal{R} \setminus \{s^N\}$.*

Proof. Perturb player i to choose action $a' > a_1$. In the next state s', because $V^{(a',a_1)} = \underline{u} + \delta Q_i^{a_1}(s') = \underline{u} + \delta u(a_1, a_1)/(1-\delta)$, we have $Q_i^{a'}(s') = Q_i^{a'}(s^N)$. Additionally, due to Assumption 1.4, $Q_{-i}^{a_1}(s')$ does not decrease. Hence, in the following states, they both continue

to play a_1, and eventually, the state returns to s^N with probability 1 ($Q_{-i}^{a_1}(s')$ converges to $u(a_1, a_1)/(1-\delta)$). □

Step 2. For all $(s, s') \in E(\underline{h}_{s^N})$, $C(s, s') = 1$. This step, the most crucial one, is accomplished by explicitly defining a strictly partial order $\prec$ over the set of all absorbing states in the next subsection. Regarding the order $\prec$, s^N is the lowest and for any absorbing state $s \in \mathcal{R}$, there always exists $s' \in \mathcal{R}$ such that $C(s, s') = 1$ and $s' \prec s$.

Step 3. Because all s-rooted spanning tree has the same number of edges and for any $s, s' \in \mathcal{R}$, $C(s, s') \geq 1$ (exiting an absorbing state requires at least one mutation). Then, the above two steps shows that $C(\underline{h}_s) \geq C(\underline{h}_{s^N}) + 1$ for all $s \in \mathcal{R} \setminus \{s^N\}$. It means that s^N is the unique stochastically stable state.

3.3 The Strict Partial Order

Remind that Proposition 1 implies that every absorbing state s has a unique symmetric action profile $(a, a) \in \mathcal{A}(s)$. For any absorbing state s, denote $a(s)$ the action chosen by both players at state s, $\underline{i}(s) \in \arg\min_{i \in I} Q_i^{a_1}(s)$ the agent with the lowest Q-value of a_1 and $\bar{i}(s) \in \arg\max_{i \in I} Q_i^{a_1}(s)$ the agent with the highest Q-value of a_1.[11] Now we formally define the order.

Definition 1. *Fix a mapping $g : A \to A$, define an order $\prec_g$ on $\mathcal{R}$ as $s \prec_g s'$ if the following conditions are met:*

1. *There exists an $\hat{a} < \underline{a}(s, s')$ such that $\forall a < \hat{a}$, $Q^a_{\underline{i}(s)}(s) = Q^a_{\underline{i}(s')}(s')$ and $Q^{\hat{a}}_{\underline{i}(s)}(s) < Q^{\hat{a}}_{\underline{i}(s')}(s')$, where $\underline{a}(s, s') := \min\{a(s), a(s')\}$.*
2. *If condition 1 does not hold, then $a(s) < a(s')$.*
3. *If condition 1 does not hold and $a(s) = a(s')$:*
   (a) *If $a(s) \neq a_1$, then $Q^{g(a(s))}_{\bar{i}(s)}(s) > Q^{g(a(s'))}_{\bar{i}(s')}(s')$.*
   (b) *If $a(s) = a_1$, then $D(s) < D(s')$, where $D(s) := \sum_{i \in I} \sum_{a \in A} |Q_i^a(s) - Q_i^a(s^N)|$.*

Two remarks are drawn on the order. First, the state s^N is the lowest, i.e. $s^N \prec_g s$ for any $s \in \mathcal{R} \setminus \{s^N\}$. The set of absorbing states where (a_1, a_1) is selected is always lower, i.e.

[11] If both agents happen to share the same Q^{a_1}, then $\underline{i}$ is assigned to the agent with the lower Q^{a_2}. If both agents also have the same Q^{a_2}, then $\underline{i}$ is assigned to the one with the lower Q^{a_3}, and so forth. If their Q-vectors, $\mathbf{Q}_1$ and $\mathbf{Q}_2$, are identical, one agent is arbitrarily designated as $\bar{i}$ and the other as $\underline{i}$.

for any s and s' in $\mathcal{R}$, if $a(s) = a_1 \neq a(s')$, then $s \prec_g s'$. Intuitively, if $s \prec_g s'$, then s is *closer* to s^N, comparing to s'. In this regard, case 2 and 3b in the definition are intuitive. Case 2 requires that the action chosen at "closer" state be strictly lower while case 3b requires that the gap in Q-values to state s^N decrease for "closer" state. The rest of the definition for the order exhibits differences in the role between the agent $\bar{i}$ with higher Q^{a_1}, and $\underline{i}$ with lower Q^{a_1}, which will be further elaborated in Subsection 5.1. On one hand, the priority of the order (i.e. case 1) is to ensure that the agent $\underline{i}$ has strictly lower Q-value for the action, the lowest one strictly below $\underline{a}(s, s')$ such that the corresponding Q-values strictly differ. On the other hand, case 3a requires that the Q-value of the agent $\bar{i}$ for a prescribed action strictly increase. The next lemma then shows that the defined order is a strict partial one.

Lemma 4. *$\prec_g$ is a strict partial order for any $g(\cdot)$, i.e, $\prec_g$ satisfies*

1. *Irreflexivity: not $s \prec_g s$.*
2. *Transitivity: if $s_1 \prec_g s_2$, $s_2 \prec_g s_3$, then $s_1 \prec_g s_3$.*

As indicated in Step 2 of the sketch of the proof, the construction of the order is to ensure that, for any absorbing state s, there always exists a single perturbation on one agent's selected action such that the resulting convergent absorbing state post the perturbation is strictly "closer" to the state s^N. The mapping $g(\cdot)$ specifies the action, to which the selected action is perturbed. The next lemma then formally prove the argument.

Lemma 5. *There exists $g(\cdot)$, such that for all $s \in \mathcal{R} \setminus s^N$, there is $s' \in \mathcal{R}$ s.t. $s' \prec_g s$ and $C(s, s') = 1$.*

Dolgopolov (2024) also explicitly construct an order on absorbing states in Prisoner's Dilemma, which requires $Q_{\bar{i}}^{a_1}$ increase and $Q_{\underline{i}}^{a_1}$ decrease for "closer" state.[12] There are two differences in constructing the order when extending to games with more than two actions. First, one has to explicitly specify the mapping g, i.e. the valid perturbation (the one which transits state closer to s^N) for each absorbing state. In Prisoner's Dilemma, the valid perturbation is simple, i.e. always perturbing to action a_1. However, in general class of games, different absorbing states and different game forms may have different sets of valid perturbations. Second, the main challenge is to address the scenario of bilateral rebound, where *an absorbing state s may transit to s' with $a(s) < a(s')$*, even under valid perturbations.

[12] Dolgopolov (2024) defines the order on cost function. However, in order to accommodate the ε-greedy exploration mode, where the cost of perturbation remains the same for different states, the order has to be defined on Q-values directly.

However, in Prisoner's Dilemma, whenever the state transits into the class $\{s : a(s) = a_1\}$, it will never leave the class given unilateral perturbation. Therefore, in our broader game framework, we cannot simply say that $s \prec s'$ if $a(s) < a(s')$. To tackle with this issue, we specify perturbations which decrease agent $\underline{i}$'s Q-value of low actions even if his $Q_{\underline{i}}^{a_1}$ does not change or converges to a higher action profile. Such perturbation diminishes the likelihood of bilateral rebound. We will elaborate it in Subsection 5.1.

4 Simulation Results

We construct two "memoryless"[13] Q-learning algorithms and let them interact in Bertrand games with consumer WTP equal to 1.[14] We focus on a baseline setting where the number of actions is $K = 10$ (resulting in action space $A = \{0.1, 0.2, \ldots, 1\}$ and $(0.1, 0.1)$ as the unique strict NE), the discount factor is $\delta = 0.95$, and the learning rate is $\alpha = 0.15$. The Q-learning algorithm uses a ε-greedy strategy with time-descending, $\varepsilon_t = e^{-\beta t}$, where $\beta = 10^{-4}$ in the baseline setting. We use the following piratical criterion to verify convergence: convergence is considered achieved if, for each player, either the optimal action does not change for 10^5 consecutive periods. We stop the algorithms if they do not converge after 1 billion periods. Each "experiment" is iterated with 100 sessions.[15]

To quantify the degree of collusion, we use the average convergent price, $\overline{p}$, as a measure of the collusive level. We consider the learning algorithms to have reached a more collusive outcome if $\overline{p}$ is higher, and a more competitive outcome if $\overline{p}$ is lower. [16]

Note that the underlying economy is of certain with a single state, therefore the Markov Decision Problem can be degenerated into a static one. In this sense, we compare outcomes of games played by standard Q-learning agents (with strictly positive discount factors) and that by degenerated Q-learning agents (with zero discount factor). To obtain relatively robust results, we explore the entire grid of a 19×21 points by varying α in a range $[0.05, 0.95]$ and β in $[10^{-5}, 5 \times 10^{-4}]$.

[13] By "memoryless", we mean that the last period price history is not taken as the state variable of Q-learning algorithms, contrast to the approach adopted by Calvano et al. (2020).

[14] Essentially, the Bertrand game and the first-price auction in section 2 are the same.

[15] Please refer to Appendix A for details on the Q-learning and Appendix C for details on simulation design.

[16] Calvano et al. (2020) and subsequent literature adopt the collusion index (CI), defined as $\text{CI} := \frac{\overline{r} - r^N}{r^M - r^N}$, where $\overline{r}$ represents the total profit upon convergence, averaged across 100 sessions, r^N is the total profit in the static NE, and r^M is the profit under monopoly. In our setting, the convergent price profile is symmetric, so $\overline{r} = \overline{p}$. Therefore, the collusion index CI is simply a normalized version of the average convergent price $\overline{p}$.

[Insert Fig.3]

The convergent price for various learning and exploration rate between standard Q-learning agents and between degenerated Q-learning agents are shown in Figure 3(a) and Figure 3(b), respectively. In Figure 3(a) where the discount factor $\delta = 0.95$, the collusive level remains remarkably high, except for the region where α approaches 1. When $\alpha \leq 0.8$, even the lowest average convergent price exceeds 0.69. In contrast, in Figure 3(b) where discount factor $\delta = 0$, the convergent outcome on average is close to NE. When $(\alpha, \beta) = (0.05, 4.5 \times 10^{-4})$, the average price reaches its maximum, i.e. 0.20. This result echoes with the simulation outcome documented by Calvano et al. (2020), who show that the (learning) algorithmic collusion will disappear if memory is deactivated. However, they only utilize simulations on games between degenerated Q-learning agents to represent the "memoryless" scenario. Therefore, it is ambiguous whether algorithmic collusion is led by strictly positive discount factor or by last period price history as state variable, based on which a reward and punishment strategy to sustain collusion is adapted to, or just by strictly positive discount factor.

[Insert Fig.4]

To further study the impact of discount factor on collusion outcome among AI agents, Figure 4 documents the average price for different discount factors. First, even without memory, strictly positive discount factor alone can induce a relatively high collusive degree. The average convergent price exceeds 0.55 (half collusive level) only when $\delta > 0.7$, even when α is less than 0.15. This poses a challenge to regulators and legist in demarcating algorithmic collusion. Harrington (2018), based on Calvano et al. (2020), proposes that algorithmic collusion can be discerned if a unilateral price undercut is followed by a price war. However, when memory is deactivated but with strictly positive discount factor, both agents will soon move back to the converging, given a unilateral price undercut. Such behavior, though can not be classified as collusion based on the criterion proposed by Harrington (2018), does harm both consumers' surplus and social welfare.[17] On one hand, strictly positive discount factor is hard to be excluded in real life where firms face more complex environment and their pricing decisions are also determined by other factors. Therefore, the Markov Decision Problem seems impossible to be degenerated by then. On the other hand, the memory of last period price history requires the knowledge of both your opponents and their actions. Such knowledge may not be consistent with the model-free nature of reinforcement learning. If

[17]Similar concern has also been raised by Banchio and Mantegazza (2023).

only payoff matrix is unknown, firms may not have to turn to reinforcement learning to make price decisions or perfectly follow the recommended price policy by artificial intelligence.[18] Second, the collusive level is significantly sensitive to variations in discount factors. The literature and even our theory part seems to be silent to the role of discount factor. Our next section studies how reinforcement learning algorithms may adapt to collusive outcomes in real simulation with insufficient learning, within which the discount factor plays the critical role.

4.1 Constant Exploration Rate

To verify predictions of our theory, we also consider simulations with constant but minute exploration rate ε. The simulation is run for a fixed number of periods $\frac{T}{\varepsilon}$ ($T \geq 10^4$). We then calculate the average price in the last $\frac{10^4}{\varepsilon}$ periods, weighted by the frequency percentage.[19] Note that $\frac{1}{\varepsilon}$ measures the length of periods, over which each Q-agent conducts exploration once in expectation. Hence, T measures the number of explorations in expectation for each Q-agent over the length $\frac{T}{\varepsilon}$ of periods. (Appendix C.4 details the method used to handle simulations with numerous periods.)

[Insert Figure 5]

Figure 5 presents convergent prices for different discount factors. Unlike outcomes of simulations with decay exploration rate, where convergent prices gradually increase with discount factors, the collusive level almost vanishes when the discount factor is below some threshold around 0.75. When the expected number T of explorations increases or the exploration rate ε decreases, this threshold increases, though slightly. Such simulation results are consistent with our theory, which demonstrates that the frequency of playing any non-NE strategy profile vanishes in the long run if the exploration rate ε is small enough.

However, when the discount factor δ is above the threshold, similar to simulation outcomes with decay exploration rate, though sharply, the convergent prices on average increase in discount factor. For instance, if $\delta > 0.8$, the convergent prices exceed 0.55 on average. Again, the discount factor significantly slows down the speed of convergence, leading to insufficient learning and therefore collusive outcome when the discount factor is sufficiently high. In that scenario, the role of discount factor can still not be silenced even if sufficient

[18] Similar criticism, perhaps more severe, have been drawn on perfect synchronous updating, introduced by Asker et al. (2022).

[19] Only symmetric prices are considered.

explorations (i.e. high T) are allowed. This is due to finite precision in the choice of exploration rate ε (constrained by precision η), computational limitations in the total length of periods T/ε and the enormous number of states (which is $O(1/\eta^K)$).

5 Channel underlying AI Collusion

The inconsistency between our theory and simulation indicates that the collusive outcomes are a result of failure to learn (strict) NE due to insufficient learning in real simulations. In this section, we propose factors which slow down the learning process toward NE.

To illustrate the underlying mechanisms for algorithmic collusion in real simulations with insufficient learning, we first introduce how reinforcement-learning agents may gradually converge to NE, based on common wisdom. Starting at some common price level, if one agent i searches for some prices p_i downward, then this agent will grab all the market share and improves his profits at that period. This boosts its Q-value of that lower action and reinforces him to keep choosing this lower action in the future. As the other agent j continuously loses market share and profits, the Q-value of the original action decreases. Once the Q-value is lower than the second largest Q-value, the other Q-agent j has to turn to other prices. For any prices strictly above p_i, its Q-value will also decrease for the same reason, which forces agent j to search for another price level. Until agent j finds some price level strict below p_i, then he will maintain his choice in the future. Now it is agent i's turn to lose market share and search for another price level. If this process, named as *sequential and alternating price undercut*, can be iterated without interruption, then these Q-learning agents are supposed to adapt to NE gradually. Besides, once they reach there, it becomes much more difficult to escape since unilateral deviation reinforces both to keep choosing NE price level.

Note that Q-values for low prices are inflated for some reasons, which will be elaborated in the next subsection. This process will be interrupted since, at low prices, even if the agent wins the market share, the profit he gathered in the period is too low to increase the Q-value. Consequently, after some time, the winning agent, together with the losing agent, must abandon this price and rebound to a higher one.

Specifically, we define the *sustainable price* as the lowest price choosing which, upon grabbing all the market share, will induce Q-learning agent to maintain his selection, instead of seeking another price level. At time t, let $Q_t^{(2)}$ be the second-highest Q-value. The

sustainable price $\underline{p}_t$ at time t is defined as

$$\underline{p}_t = (1-\delta)Q_t^{(2)}. \tag{10}$$

whenever the winning Q-agent undercut his price p strictly below $\underline{p}_t$, the Q-value of price p will be strictly lower than $Q_t^{(2)}$ in the long run. Based on the decision rule of Q-learning, the winning Q-agent, similar to the losing agent, then have to choose another price, which is more likely to be a higher one. Such phenomenon, where both agents rebound to higher prices, is termed *bilateral rebound*. Note that bilateral rebound does not require agents to act simultaneously. If they do not rebound to the same price level, then the process of sequential and alternating price undercut, together with bilateral rebound, will be iterated, making the price curves look like an *Edgeworth cycle* as shown in Figure 6.

[Insert Figure 6]

However, if both Q-agents coincidentally rebound to the same price level, which is relatively high, then they may get stuck there in real simulations. Specifically, we define the *stationary price* as the lowest price choosing which will induce both Q-agents maintain their choices if both share the market. Specifically, the stationary price $\hat{p}_t$ is defined as

$$\hat{p}_t = 2(1-\delta)\max\{Q_{i,t}^{(2)}, Q_{j,t}^{(2)}\}. \tag{11}$$

Whenever both Q-agents rebound to a price p strictly higher than the stationary price $\hat{p}_t$. Their Q-value at price p will keep increasing and converge to $\frac{p}{2(1-\delta)}$. There is a strictly positive gap between their Q-value at price p and their second highest Q-value. Such gap can resist to unilateral deviations within a given period, so long as the exploration rate is diminishing to a low enough scale.

Q-value Inflation and Discount Factor. Remind that Q-values at $t = 0$ are initialized at the discounted payoff that would accrue to player i if his opponents randomized uniformly. Therefore, initial Q-values are in a plausible range that it reinforces any action whenever it wins all the market share. One question left unanswered in the last subsection is where *inflation* of Q-values for low prices comes from.

Return to the updating rule of Q-learning,

$$Q_i^{a_i}(s_{t+1}) = (1-\alpha)Q_i^{a_i}(s_t) + \alpha\left[u(a_i, a_{-i}) + \delta \max_{a' \in A} Q_i^{a'}(s_t)\right], \forall i \in I. \tag{12}$$

Note that, in early periods, actions are mainly chosen due to extensive exploration. Its Q-values are updated by injecting the maximal Q-value and therefore inflated. Later on, whenever an exploration happens, its Q-value will then be inflated. Figure 7 documents one sample of Q-values in early periods. The Q-values for different actions, even if they are initialized at different values, always converge to same value in early phases. The main reason is that, in early phases, actions are randomly chosen and Q-values are "injected" with same maximal Q-value. Besides, the convergent Q-values are around 6.5.

[Insert Figure 7]

To ensure that the process of sequential and alternating price undercut will not be interrupted, one approach is to increase the minimal price. Even if the Q-values are inflated, the minimal action is still sustainable, excluding the possibility of bilateral rebound. Figure 8 shows how the average convergent price responds to variations in minimal actions. If minimal actions are relatively low, increasing the minimal action can effectively decrease the convergent price and therefore the collusive level. Beyond the threshold of around 0.33, the convergent price is almost the minimal action, where the collusion vanishes. Note that the threshold 0.33 is approximately the sustainable price for 6.5, around which early Q-values for each prices converge. One policy recommendation, allowing for mild collusion, through regulating firms to improve the minimal price it can charge, can avoid relatively high collusion.

[Insert Figure 8]

In this regard, given strictly positive discount factor, exploration acts as a sword with double edges. On one hand, it initiates price undercut and prevent stuck at high prices. On the other hand, it inflates the Q-value, interrupting the process of price undercut.

5.1 Relation to the Theory

Remind that, in the theory part, the key step for the proof is, for each state (i.e. pair of vectors of Q-values), to explicitly construct a path of perturbations toward the state s^N, on which (strict) NE profile is played. Along the path, only unilateral mutation is required to transit from one absorbing state to the next. Based on the order we explicitly proposed, such unilateral mutation is required to either decrease Q-values of low actions for agent $\underline{i}$ (case 1) or increase Q-values of the prescribed actions for agent $\bar{i}$ (case 3a). Note that bilateral

rebound is allowed in case 1, i.e. the selected actions may be strictly higher than that in previous state. The decrease of Q-values for low actions makes it more likely for these low actions to be sustainable, when they are chosen during the process of price undercut. When case 3a is realized, the increase of Q-values for prescribed actions makes it more likely to initiate the process of price undercut. Therefore, the stark difference between agent $\bar{i}$ and $\underline{i}$ reflects their different roles in the process of alternating and sequential price undercut. On one hand, agent $\bar{i}$ is supposed to initiate the process, therefore higher Q-value decreases the likelihood of price stuck, facilitating downward price search. On the other hand, agent $\underline{i}$ is supposed to sustain the process, therefore lower Q-value decreases the sustainable line, facilitating further price undercut and diminishing the likelihood of bilateral rebound.

However, in real simulations, perturbations are conducted randomly and some invalid perturbations may move the current state away from the state s^N. For instance, if agent $\underline{i}$ conducts explorations by choosing some low actions and then reverts back to the original actions on path, then the Q-value for the chosen low action will be inflated and the resulting new state is further away from s^N. As the number of states grows, the probability of following the prescribed perturbations consecutively at each state diminishes exponentially. Therefore, with strictly positive probability, the process will be trapped in cycles in finite periods. Such cycles, the probability of which vanishes as the length of periods increases (with constant and minute exploration), do decrease the converging speed and lead to prices being stuck at relatively high level, in real simulations with insufficient learning.

Final remark is drawn on the classical result on the speed of convergence as $\varepsilon \to 0$, developed by Ellison (2000). Note that the discount factor is completely muted in his measure of converging speed. Given the computational precision limits, the exploration rate ε cannot be set small enough relative to the vast (though still finite) number of states. Thus, the observed convergence speed given $\varepsilon > 0$ is a more practical explanation for the simulated outcomes. Unfortunately, analyzing this property in detail, except in limit cases, is extremely difficult in mathematics. Consequently, we will focus on analyzing the channels through which the discount factor affects the convergence speed using simulation analysis.

6 Applications

6.1 Optimistic Initialization

Asker et al. (2022, 2023) show that even when $\delta = 0$, Q-learning with *optimistic initialization*

will achieve a significantly high collusive level in Bertrand games. Optimistic initialization is a technique used in reinforcement learning to encourage exploration by initializing the Q-values (or value estimates) of actions to higher values than they would typically start with (see footnote 21). The idea is that by starting with high Q-values, the learning algorithm is more likely to explore different actions, as the initial estimates suggest these actions might yield high rewards.

Calvano et al. (2023) show that this result is not robust, when exploration induced by optimistic initialization is replaced by (or combined with) ε-greedy, the collusion disappears. This finding aligns with our analysis.

[Insert Figure 9]

The reason why optimistic initialization induces a high collusive outcome even without a discount factor can be easily interpreted using the concept of the sustainable line. With very high initial Q-values, the sustainable line is also very high initially, making all actions unsustainable (stage A in Figure 9). As exploration continues, Q-values gradually decrease. During this period, higher actions become sustainable first and subsequently become stationary (stage B in Figure 9). Before lower prices can become sustainable, agents will bilaterally rebound to higher actions repeatedly. Consequently, there is a significant probability that they will bilaterally rebound to the same high price, at which point they will remain there and stop learning (stage C in Figure 9).

6.2 Market Design

As shown in Banchio and Skrzypacz (2022), when auctions are played by two learning algorithms, they converge to different outcomes depending on the auction type. In first-price auctions (FPA), they tend to converge to a collusive outcome (almost the lowest bid), while in second-price auctions (SPA), they converge to the strict NE (the highest bid). Banchio and Skrzypacz (2022) identified two forces in FPA: one that directs bids upwards and another that steers them downwards, while in SPA, the latter force disappears. We will provide a more detailed explanation of this phenomenon. The key difference between FPA and SPA is that in FPA, the sustainability is based on the agent's own action, whereas in SPA, it is based on the opponent's action.

In FPA, if an agent's action is higher than the sustainable line, she will eventually rebound to a lower bid regardless of the opponent's bid. Therefore, in FPA, when bids

increase, even if one agent first rebounds to a lower bid due to a decrease in their Q-value, the other will also rebound in the future (though not necessarily at the same time). We term this behavior *bilateral rebound*. The following scenario aligns precisely with our analysis in the baseline environment.

[Insert Figure 10]

Conversely, in SPA, when agents compete with higher bids, if one agent rebounds to a lower bid (below the sustainable line of the other agent), the other agent has no incentive to rebound (stage A and B in Figure 10). We call this *unilateral rebound*. Over time, because one agent remains at the higher bid in SPA, the agent who rebounded to the lower bid will eventually return to the higher range. When she returns, her sustainable price will decrease, as the second-largest Q-value decreases with each switch in action. Thus, their sustainable prices will gradually and alternately decrease until they are higher than the highest bid. In subsequent periods, at least one agent will remain at the Nash bid, squeezing the bubble in the Q-values (stage C in Figure 10). Eventually, the stationary lines will stabilize above the Nash bid, and the agents will converge to the NE.

When a player's profit depends on the opponent's action, the transition from bilateral rebound to unilateral rebound ultimately occurs, leading to convergence to the NE instead of a collusive outcome. We further analyze the outcome of a convex combination of FPA and SPA, where a player's profit partially depends on both their own action and their opponent's action. In this combined auction, we cannot eliminate bilateral rebound but can reduce its probability while increasing the probability of unilateral rebound, thereby enhancing the likelihood of convergence to the NE.

[Insert Figure 11]

Back to Example 2, consider a liner combination of FPA and SPA, denoted by ω FPA $+ (1-\omega)$ SPA. Figure 11 shows that when the weight on FPA is less than 0.45, the bids almost always converge to the highest bid. The theory part may provide some intuition for the threshold 0.45. The essence of the constructed order in Definition 1 is, for each absorbing state s, to specify a sequence of valid and unilateral perturbations, which move s to s^N step by step. Fix an absorbing state, the set of valid perturbations is sensitive to the underlying game form. In SPA, unilateral perturbations to any bid strictly above the selected one are valid. In FPA, however, only unilateral perturbations to slightly higher bids can accommodate orders we proposed. Otherwise, the perturbation may transit to either a

"further" state or an incomparable state. If FPA has strictly less perturbation paths toward s^N than SPA, then the probability of converging to the strict NE in FPA within a given period should be strictly lower. The following proposition explicitly characterizes the set of valid perturbations (which can accommodate our order and the rigorous definition is in Appendix B.6) for each convex combination of FPA and SPA.

Proposition 2. *When $\delta > 0$ and consider auction ω FPA + $(1-\omega)$ SPA, for absorbing state with selected bid $b \in A \setminus \left\{\frac{(K-1)v}{v}\right\}$, the set of valid perturbations $g(\cdot)$ satisfies*

$$\{g(b)\} = \begin{cases} \left(b, \frac{Kv}{K+1}\right] & \text{if } \omega \leq \frac{(K-1)v+(K+1)b}{2[Kv+(K+1)b]}, \\ \left(b, \frac{v+(2\omega-1)b}{2(1-\omega)}\right] & \text{otherwise.} \end{cases} \tag{13}$$

For every bid $b \in B$, if $g(b)$ can be arbitrarily selected among $b' > b$, $\omega \leq \min_b \frac{(K-1)v+(K+1)b}{2[Kv+(K+1)b]} = \frac{K-1}{2K}$. When $K = 10$, this implies $\omega \leq 0.45$. It means that when the weight on FPA is lower than 0.45, the set of valid perturbations is exactly the same as that of SPA, which may result in a similar speed of convergence.

6.3 Algorithm Design

Asker et al. (2022) discuss the differences between asynchronous updating, (perfect) synchronous updating, and synchronous updating using downward demand. Classic Q-learning employs asynchronous updating, where in each period the algorithm updates only the Q-value of the action taken.

In asynchronous updating, the algorithm updates the Q-value of the chosen action based on the actual return obtained, without considering counterfactual outcomes.

In synchronous updating, the algorithm updates not only the Q-value of the action taken but also the Q-values of all other actions using the counterfactual return that could have been obtained if the agent had played another action, holding the opponent's action fixed. For all $i \in I$, when the opponent chooses action a^{-i} at period t, all Q-values are updated as follows:

$$Q_i^a(s_{t+1}) = (1-\alpha)Q_i^a(s_t) + \alpha\left[u(a, a^{-i}) + \delta \max_{a'} Q_i^{a'}(s_t)\right], \forall a \in A. \tag{14}$$

In synchronous updating using downward demand, the algorithm knows the demand it receives each period, denoted by $q(a^i, a^{-i})$. The upper (lower) bound of profit for another

action is given by $\tilde{u}(a|a^i, a^{-i}) = (a - c)q(a^i, a^{-i})$ for $a > a^i$ (for $a < a^i$), where c is the marginal cost. Then all Q-values are updated as follows:

$$Q_i^a(s_{t+1}) = \begin{cases} \min\left\{Q_i^a(s_t), (1-\alpha)Q_i^a(s_t) + \alpha\left[\tilde{u}(a|a^i, a^{-i}) + \delta \max_{a'} Q_i^{a'}(s_t)\right]\right\} & \text{if } a > a^i, \\ (1-\alpha)Q_i^a(s_t) + \alpha\left[u(a^i, a^{-i}) + \delta \max_{a'} Q_i^{a'}(s_t)\right] & \text{if } a = a^i, \\ \max\left\{Q_i^a(s_t), (1-\alpha)Q_i^a(s_t) + \alpha\left[\tilde{u}(a|a^i, a^{-i}) + \delta \max_{a'} Q_i^{a'}(s_t)\right]\right\} & \text{if } a < a^i. \end{cases} \tag{15}$$

Asker et al. (2022) show that in Bertrand competition, asynchronous updating converges to a very high price, while perfect synchronous updating converges to almost the lowest price, and synchronous updating using downward demand converges to near the lowest price. Although Asker et al. (2022) obtained these results with a discount factor $\delta = 0$, Banchio and Skrzypacz (2022) corroborate these findings in the environment of FPA with $\delta > 0$.

The reason why synchronous updating converges to the NE is straightforward. When the learning algorithm can make counterfactual estimations, the action equal to (or slightly lower than) the best response is always sustainable, preventing rebounds (i.e., the sustainable line will always be lower than the best response). During the downward search phase, the agent is more likely to choose the best response because the corresponding Q-value increases at the fastest rate. Thus, sequential and alternating downward searches continue until they converge to NE.

Synchronous updating using downward demand provides an upper bound for Q-values corresponding to high prices and a lower bound for Q-values corresponding to low prices. Consequently, the sustainable line is lower than in asynchronous updating, making bilateral rebounds less likely to occur and increasing the likelihood of reaching the NE outcome.

7 Discussion on "Memory" and Future Work

In this paper, we do not formally analyze the scenario where the learning algorithm has a one-period memory of previous prices. It remains unclear whether such memory would enable the algorithm to adopt a trigger strategy and sustain genuine collusion, as suggested by Calvano et al. (2020, 2023). In Calvano et al. (2020), where algorithms lacked memory, the discount factor was set to zero by default, leading the algorithm to converge to a competitive outcome. Figure 12 illustrates the relationship between convergent prices and the

discount factor. Generally, as the discount factor increases, the pattern of collusive behavior closely resembles that observed in the memoryless case, indicating that the discount factor significantly influences collusion outcomes. When considering only outcomes without cycles, the collusive level is more closely aligned with that of the memoryless scenario and is strictly lower than the collusive level observed with memory. This raises the question: what is the net effect of removing the discount factor versus introducing memory on collusion outcomes?

Compared to memoryless learning, one-period memory increases the price when the discount factor is below 0.65 but decreases it when the discount factor exceeds 0.65. Introducing memory could lead to cycles, making the convergent price more variable and rendering convergence to the NE more challenging (see Figure 13). However, it remains an open question whether introducing memory will cause the Markov process to develop other stochastically stable states besides the NE, where off-path actions involve punishment strategies, or if these punishment strategies are merely the result of insufficient learning, reflecting the path of sequential undercutting and rebounding. The specific impact of incorporating price history remains unclear, and understanding this effect is an important area for future research.

[Insert Figure 12 and 13]

References

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

Asker, J., C. Fershtman, and A. Pakes (2022). Artificial intelligence, algorithm design, and pricing. In *AEA Papers and Proceedings*, Volume 112, pp. 452–56.

Asker, J., C. Fershtman, and A. Pakes (2023). The impact of artificial intelligence design on pricing. *Journal of Economics & Management Strategy*.

Banchio, M. and G. Mantegazza (2023). Adaptive algorithms and collusion via coupling. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC '23, New York, NY, USA, pp. 208. Association for Computing Machinery.

Banchio, M. and A. Skrzypacz (2022). Artificial intelligence and auction design. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, New York, NY, USA, pp. 30–31. Association for Computing Machinery.

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2023). Algorithmic collusion: Genuine or spurious? *International Journal of Industrial Organization 90*, 102973.

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

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2021). Algorithmic collusion with imperfect monitoring. *International Journal of Industrial Organization 79*, 102712.

Colliard, J.-E., T. Foucault, and S. Lovo (2022). Algorithmic pricing and liquidity in securities markets. *HEC Paris Research Paper*.

Cui, Z. and J. Zhai (2010). Escape dynamics and equilibria selection by iterative cycle decomposition. *Journal of Mathematical Economics 46*(6), 1015–1029.

Dolgopolov, A. (2024). Reinforcement learning in a prisoner's dilemma. *Games and Economic Behavior 144*, 84–103.

Dou, W. W., I. Goldstein, and Y. Ji (2023). Ai-powered trading, algorithmic collusion, and price efficiency. *Available at SSRN 4452704*.

Ellison, G. (2000). Basins of attraction, long-run stochastic stability, and the speed of step-by-step evolution. *The Review of Economic Studies 67*(1), 17–45.

Fershtman, C. and A. Pakes (2012). Dynamic games with asymmetric information: A framework for empirical work. *The Quarterly Journal of Economics 127*(4), 1611–1661.

Foster, D. and P. Young (1990). Stochastic evolutionary game dynamics. *Theoretical Population Biology 38*(2), 219–232.

Freidlin, M. I. and A. D. Wentzell (1984). *Random Perturbations of Dynamical Systems*. Springer New York, NY.

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

Jackson, M. O. and A. Watts (2002). The evolution of social and economic networks. *Journal of economic theory 106*(2), 265–295.

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

Kandori, M., G. J. Mailath, and R. Rob (1993). Learning, mutation, and long run equilibria in games. *Econometrica: Journal of the Econometric Society*, 29–56.

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

Levine, D. K. and S. Modica (2016). Dynamics in stochastic evolutionary models. *Theoretical Economics 11*(1), 89–131.

Staudigl, M. and S. Weidenholzer (2014). Constrained interactions and social coordination. *Journal of Economic Theory 152*, 41–63.

Waltman, L. and U. Kaymak (2007). A theoretical analysis of cooperative behavior in multi-agent q-learning. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 84–91. IEEE.

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

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

Werner, T. (2023). Algorithmic and human collusion. *Available at SSRN 3960738*.

Wu, J. X., Y. Wu, K.-Y. Chen, and L. Hua (2023). Building socially intelligent ai systems: Evidence from the trust game using artificial agents with deep learning. *Management Science 69*(12), 7236–7252.

Xu, Z., M. Zhang, and W. Zhao (2024). Algorithmic collusion and price discrimination: The over-usage of data. *arXiv preprint arXiv:2403.06150*.

Young, H. P. (1993). The evolution of conventions. *Econometrica: Journal of the Econometric Society*, 57–84.

A Q-Learning

Q-learning is a model-free, off-policy reinforcement learning algorithm used to find the optimal action-selection policy for a given finite Markov decision process (MDP).

In a stationary MDP, in each period an agent observes a state (of the world) $\omega \in \Omega$ and then chooses an action $a \in A$. For any ω and a, the agent obtains a reward u, and the system moves to the next state ω', according to a time-invariant (and possibly degenerate) probability distribution $P(\omega', u|\omega, a)$. The goal of the agent is to find an optimal strategy $\sigma^* : \Omega \to A$ which maximizes the agent's cumulative payoff.

Q-learning was proposed to find an optimal strategy with no prior knowledge of the underlying model, i.e., the distribution $P(\omega', u|\omega, a)$. The algorithm works by iteratively updating an action-value function $Q(a, \omega)$, which estimates the expected cumulative reward of taking action a in state ω. [20] The key idea behind Q-learning is the Bellman equation:

$$Q(a, \omega) = \sum_{\omega', u} P(\omega', u|\omega, a) \left[u + \delta \max_{a' \in A} Q(a', \omega') \right], \tag{16}$$

where $0 \leq \delta < 1$ denotes the discount factor. If Q-matrix is known, an optimal strategy is given by

$$\sigma^*(\omega) = \arg\max_{a \in A} Q^a(\omega). \tag{17}$$

Q-learning algorithm estimate the Q-matrix iteratively. Starting from an arbitrary initial matrix Q_0, after choosing action a in state ω, the algorithm observes immediate payoff u and next state ω' and updates the Q-values using the update rule:

$$Q(a, \omega) = (1 - \alpha) Q(\omega, a) + \alpha \left[u + \delta \max_{a' \in A} Q(\omega', a') \right], \tag{18}$$

where $0 < \alpha \leq 1$ is called the *learning rate*. Note that only for the cell (ω, a) visited, the corresponding Q-value $Q(\omega, a)$ is updated. It is proven in Watkins and Dayan (1992) that values $Q(\omega, a)$ estimated will converge with probability 1 to the optimal ones if each action is executed in each state an infinite number of times and α is decayed appropriately.

To approximate the true Q-matrix starting from an arbitrary $\mathbf{Q}_0$, it is essential that

[20] In the main paper, s_t represents the state of the Markov process, which is a profile of the Q-matrix. For player i, action a, and state of the world ω, the corresponding Q-value in state s_t is denoted by $Q_i^{a,\omega}(s_t)$, which is commonly referred to in the literature as $Q_{i,t}(a, \omega)$. If the set of states of the world is a singleton, it can be simplified to $Q_i^a(s_t)$.

all actions are tried in all states. Q-learning decision-making involves two key concepts: exploitation (greedy policy) and exploration (perturbation). Exploitation focuses on leveraging the agent's existing knowledge to maximize immediate rewards, while exploration allows the agent to discover new aspects of the environment and learn about unknown states and actions. Strategies (policies) such as ε-greedy, Boltzmann exploration, and upper confidence bound are commonly used in various reinforcement learning algorithms to manage this balance effectively.

B Omitted Proof

B.1 Proof of Lemma 1

The statements in Lemma 1 are derived from Lemma 2 in Dolgopolov (2024). The proof follows the same approach as in Dolgopolov (2024).

Suppose that $\mathcal{A}_i(S)$ contains at least two actions (otherwise, the case is trivial). There exists a state s_0 where player i initially plays $a' \in A$. In the next state s_1, player i transitions to action $a \in A \setminus \{a'\}$ for a duration of $\tau - 1$ units of time, and then switches to another action $a'' \in A \setminus \{a\}$ at time τ. This process is illustrated in the first sequence in Table 1. First, we will demonstrate that $Q_i^a(s_1) = Q_i^a(s_\tau)$. In other words, the Q-value of action a at the time it is first chosen is the same as its Q-value after the last time it is chosen.

state	s_0	s_1	$\dots$	$s_{\tau-1}$	s_τ	$\dots$	s'_0	s'_1	$\dots$	$s'_{\tau'-1}$	$s'_{\tau'}$	$\dots$	s_0	s_1	$\dots$
i	a'	a	$\dots$	a	a''	$\dots$	a'''	a	$\dots$	a	a''''	$\dots$	a'	a	$\dots$

Table 1: Sequences of Action Transition

Since $Q_i^{a''}(s_1) = Q_i^{a''}(s_\tau)$ (because $Q_i^{a''}$ does not update during this process), and given that $Q_i^a(s_1) \geq Q_i^{a''}(s_1)$ (as $a \in \mathcal{A}_i(s_1)$) and $Q_i^{a''}(s_\tau) \leq Q_i^{a''}(s_\tau)$ (as $a'' \in \mathcal{A}_i(s_\tau)$), it follows that $Q_i^a(s_1) \geq Q_i^a(s_\tau)$. This implies that for any chosen action, when the player transitions to another action, the corresponding Q-value is weakly lower than the value at the time it was first chosen. Suppose $Q_i^a(s_1) \neq Q_i^a(s_\tau)$, it must be that $Q_i^a(s_1) > Q_i^a(s_\tau)$.

Since s_0 is a recurrent state, there is a positive probability of returning to s_0 and subsequently transitioning to state s_1 in the future (as shown in the last sequence in Table 1). Note that s_1 is the *first* state choose a with $Q_i^a(s_1) > Q_i^a(s_\tau)$. Hence, between s_τ (the first time) and s_1 (the second time), there must exist a sequence $s'_0, s'_1, \dots, s'_{\tau'}$ (as shown in the

middle sequence in Table 1) where action a is played in all states $s'_1, \ldots, s'_{\tau'-1}$ but not played in s'_0 and $s'_{\tau'}$, and $Q_i^a(s'_{\tau'}) > Q_i^a(s'_1)$. This means that after being chosen in this sequence, Q_i^a must strictly increase. This contradicts the statement in the previous paragraph.

Given that $Q_i^a(s_1) \geq Q_i^{a''}(s_1) = Q_i^{a''}(s_\tau) \geq Q_i^a(s_\tau)$, it follows that $Q_i^a(s_\tau) = Q_i^{a''}(s_\tau)$. Note that in state s_τ, action a'' is chosen instead of a. For player i, there must exist a transition sequence $a_{k_1}, a_{k_2}, \ldots, a_{k_L}$ that includes all actions in $\mathcal{A}_i(S)$ (with actions allowed to repeat). Player i first plays a_{k_1}, then transitions to a_{k_2}, and so forth. Then $Q_i^{a_{k_{l-1}}} = Q_i^{a_{k_l}}$ at the state choosing a_{k_l} instead of $a_{k_{l-1}}$. Since the Q-value corresponding to non-chosen action will maintain, then we prove that there exists a value $\underline{q}^i$, for any $a^i \in \mathcal{A}_i(S)$, if it is not chosen, then $Q_i^{a^i} = \underline{q}^i$.

This implies that an action transition occurs only when the chosen Q-value equals $\underline{q}^i$. Thus, when the chosen Q-value is $\underline{q}^i$, player i has a positive probability of selecting any action $a^i \in \mathcal{A}_i(S)$, meaning there is also a positive probability that the player will continue with their previous action. Consequently, for all $(a^i, a^{-i}) \in \mathcal{A}(S)$, if these actions are played for a sufficiently long period, there exists a corresponding state $s \in S$ where $Q_i^{a^i}(s) = \frac{u(a^i, a^{-i})}{1-\delta}$ and $Q_{-i}^{a^{-i}}(s) = \frac{u(a^{-i}, a^i)}{1-\delta}$.

Then, because $|\mathcal{A}(S)| \geq 2$, when the state reaches s, there must be a positive probability of transitioning to another state s' where a different action profile is played. Consequently, at least one of the following equations must hold: $\underline{q}^i = Q_i^{a^i}(s)$ or $\underline{q}^{-i} = Q_{-i}^{a^{-i}}(s)$.

B.2 Proof of Proposition 1

We first prove the following two claims.

Claim 1. *Any off-diagonal action profile cannot be part of the path in any recurrent class. In other words, for any $S \in \mathcal{R}$, there does not exist an action profile $(a^i, a^{-i}) \in \mathcal{A}(S)$ where $a^i \neq a^{-i}$.*

Proof. Suppose that in $S \in \mathcal{R}$, there exists an action profile $(a^i, a^{-i}) \in \mathcal{A}(S)$ where $a^i > a^{-i}$. Since (a^i, a^{-i}) can be continuously selected for an extended period, there exists a state $s \in S$ where $(a^i, a^{-i}) \in \mathcal{A}(s)$ and $Q_i^{a^i}(s) = \frac{u}{1-\delta}$. Given that $Q_i^{a'}(s) \geq \frac{u}{1-\delta}$ for all actions a', player i has a positive probability of selecting any action, i.e., $\mathcal{A}_i(s) = A$. Additionally, according to Lemma 1, $\underline{q}^i = \frac{u}{1-\delta}$. Now, consider player i transitioning to select the lowest action, a_1. Two cases arise:

Case 1: $a^{-i} > a_1$. In this scenario, there exists another state $s' \in S$ where $(a_1, a^{-i}) \in$

$\mathcal{A}(s')$ and $Q_{-i}^{a^{-i}}(s') = \frac{\underline{u}}{1-\delta}$. Therefore, $\underline{q}^{-i} = \frac{\underline{u}}{1-\delta}$, meaning that player $-i$ can select any action in the next period. Consequently, there is a positive probability that both players will select a_1. The state can then transition to $s'' \in S$ where $Q_i^{a_1}(s'') = Q_{-i}^{a_1}(s'') = \frac{u(a_1,a_1)}{1-\delta} > \underline{q}^i = \underline{q}^{-i}$. Once they reach state s'', they cannot leave it, which contradicts the assumption that $s, s' \in S$ and S is a recurrent class.

Case 2: $a^{-i} = a_1$. In this case, both players will eventually transition to the state $s'' \in S$ as defined above. Since $s \in S$, they must return to state s. However, because $Q_i^{a_1}(s'') = \frac{u(a_1,a_1)}{1-\delta} > \underline{q}^i$, player i will continue to choose a_1 and will not change actions. Therefore, player $-i$ must have some positive probability of changing her action. This implies that $\underline{q}^{-i} = \frac{u(a_1,a_1)}{1-\delta}$. However, when player i plays a_1, player $-i$ will always receive the lowest profit, $\underline{u}$, if she plays any action other than a_1. Thus, once player $-i$ selects any action $a' \in \mathcal{A}_{-i}(s'') \setminus \{a_1\} \subset \mathcal{A}_{-i}(S)$, the state will transition to some state s''' where $Q_{-i}^{a'}(s''') < \underline{q}^{-i}$, which contradicts Lemma 1. □

Claim 2. *Any on-diagonal action profile cannot be on the path within the same recurrent class. In other words, for any $S \in R$, $\mathcal{A}(S) = \{(a, a)\}$ for some $a \in A$.*

Proof. Suppose there exists a recurrent class $S \in \mathcal{R}$ where $(a, a), (a', a') \in \mathcal{A}(S)$ with $a \neq a'$. Then, there exists a state $s \in S$ where $(a, a) \in \mathcal{A}(s)$ and $Q_i^a(s) = Q_{-i}^a(s) = \frac{u(a,a)}{1-\delta}$, and another state $s' \in S$ where $(a', a') \in \mathcal{A}(s')$ and $Q_i^{a'}(s') = Q_{-i}^{a'}(s') = \frac{u(a',a')}{1-\delta}$. Since $u(a, a) \neq u(a', a')$, it follows that $s \neq s'$.

Given Claim 1, any off-diagonal action profile cannot be on the path within S. Therefore, for the transition from state s to state s', both players must simultaneously change their actions. This implies that $\underline{q}^i = \underline{q}^{-i} = \frac{u(a,a)}{1-\delta}$. Conversely, for the transition from state s' to state s, both players again need to change their actions at the same time, which implies $\underline{q}^i = \underline{q}^{-i} = \frac{u(a',a')}{1-\delta}$. This leads to a contradiction, as it cannot be the case that $\underline{q}^i$ equals both $\frac{u(a,a)}{1-\delta}$ and $\frac{u(a',a')}{1-\delta}$ simultaneously. □

According to Claim 2, for any $S \in \mathcal{R}$, $\mathcal{A}(S) = \{(a, a)\}$ for some $a \in A$. When players continuously select a, the state will eventually converge to a state s where, for all $i \in I$, $Q_i^a(s) = \frac{u(a,a)}{1-\delta}$ and $Q_i^{a'}(s) < Q_i^a(s)$ for all $a' \in A \setminus \{a\}$. Furthermore, once the unperturbed process reaches state s, it becomes absorbing, meaning $S = \{s\}$.

B.3 Proof of Lemma 2

In Lemma 5, we demonstrate that every state $s \in \mathcal{S}\backslash\{s^N\}$ has a path leading to s^N (as defined in Equation (7)). This implies that s^N must belong to a recurrent class, suggesting that the perturbed Markov process has a unique recurrent class, denoted by $\mathcal{R}_\varepsilon$. Consequently, this ensures a unique stationary distribution, μ_ε.

The perturbed Markov process, when restricted to the subset of states in $\mathcal{R}_\varepsilon$ and denoted by $\tilde{P}_\varepsilon$, is irreducible. Consequently, the stationary distribution of $\tilde{P}_\varepsilon$, denoted by $\tilde{\mu}_\varepsilon$, can be computed using Lemma 3.1 from Freidlin and Wentzell (1984) (or Lemma 1 in Kandori et al. (1993)), where it is represented as the product of specific entries of $\tilde{P}_\varepsilon$ via the spanning tree method. Thus, the convergence properties of $\tilde{\mu}_\varepsilon$ can be derived from the convergence of $\tilde{P}_\varepsilon$. The full stationary distribution μ_ε can then be obtained by extending $\tilde{\mu}_\varepsilon$ by setting the entries outside $\mathcal{R}_\varepsilon$ to 0. Therefore, μ_ε is convergent.

In the main paper, when discussing the stochastically stable state, we focus solely on the states within $\mathcal{R}_\varepsilon$ and construct a spanning tree among these states to compute $\tilde{\mu}_\varepsilon$. For simplicity of presentation, we do not emphasize this distinction rigorously.

B.4 Proof of Lemma 4

We only prove the transitivity of $\prec_g$. First, suppose $s \prec_g s'$ due to condition 1, with the threshold $\hat{a}(s, s')$. Regardless of the condition under which $s' \prec_g s''$ holds, we have $s \prec_g s''$ due to condition 1: If $s' \prec_g s''$ is also due to condition 1, with threshold $\hat{a}(s', s'')$, then since $\underline{a}(s, s'') \geq \min\{\underline{a}(s, s'), \underline{a}(s', s'')\}$, it follows that $\min\{\hat{a}(s, s'), \hat{a}(s', s'')\} < \underline{a}(s, s'')$ is a valid threshold satisfying condition 1 for $s \prec_g s''$. Otherwise, $\underline{a}(s, s'') \geq \underline{a}(s, s')$ since $a(s') \leq a(s'')$, then $\hat{a}(s, s') < \underline{a}(s, s'')$ also serves as a valid threshold satisfying condition 1 for $s \prec_g s''$.

Second, suppose $s \prec_g s'$ is determined by condition 2. If $s' \prec_g s''$ is due to condition 2 or condition 3, then it is straightforward that $s \prec_g s''$ holds by condition 2. If $s' \prec_g s''$ is due to condition 1 with threshold $\hat{a}(s', s'')$: If $\hat{a}(s', s'') < a(s)$, then $\hat{a}(s', s'') < \underline{a}(s, s'')$ is also a valid threshold satisfying condition 1 for $s \prec_g s''$; otherwise, because $a(s) \leq \hat{a}(s', s'') < a(s'')$, $s \prec_g s''$ holds by condition 2.

Finally, consider the case where $s \prec_g s'$ is determined by condition 3. No matter $s' \prec_g s''$ is governed by which condition, then $s \prec_g s''$ will also hold under the same condition. This is because $a(s) = a(s')$ ensures the transitivity of condition 1 and 2, while condition 3 is inherently transitive.

B.5 Proof of Lemma 5

For any $s \neq s^N$, if $a(s) = a_1$, then there must be an action $a' \neq a_1$ such that $Q_i^{a'}(s) \neq Q_i^{a'}(s^N) = \underline{u} + \delta\,[u(a_1, a_1)/(1-\delta)]$. If we perturb player i to choose a', in the subsequent state s'', for player $-i$, $Q_{-i}^{a_1}(s'')$ increases while other Q-values remain unchanged; for player i, $Q_i^{a'}(s'')$ becomes closer to $Q_i^{a'}(s^N)$ but remains lower than $Q_i^{a_1}(s'')$. Hence, both players will always choose a_1 in the subsequent states, and the state will eventually converge to s', where $Q_i^{a'}(s')$ is closer to $Q_i^{a'}(s^N)$ while other values remain unchanged. According to condition 3b, $s' \prec_g s$ for any g.

In the following, suppose that $a(s) \neq a_1$. According to Assumption 1.3, there exists $a' < a(s)$ such that $u(a', a(s)) \geq u(a(s), a(s))$. We set $g(a(s)) = a'$ hereafter (if there are multiple suitable a', we arbitrarily select one). Then, perturb player $\bar{i}(s)$ to choose a', and the state will eventually converge to s'. If $a(s') = a_1$, then $s' \prec_g s$. So, w.l.o.g., we assume $a(s') \neq a_1$.

To proceed with the proof, we first rigorously demonstrate that if the lowest action chosen by a player is not been selected greedily in the next absorbing state, then the Q-value of this lowest action will *strictly* decrease with a positive probability.

Lemma 6. *At state $s \in \mathcal{R}$, after perturbing player $\bar{i}(s)$ to choose $a' \neq a(s)$, there is a positive probability that the next absorbing state s' will either have $a(s') = \underline{a}_{\underline{i}(s)}$, the lowest action chosen by player $\underline{i}(s)$ in the process from s to s', or $Q_{\underline{i}(s)}^{\underline{a}_{\underline{i}(s)}}(s') < Q_{\underline{i}(s)}^{\underline{a}_{\underline{i}(s)}}(s)$.*

Proof. Consider a modified unperturbed Markov process where, if player $\underline{i}(s)$ can select from more than two actions at a given time, she will always choose the lowest one with probability 1.

In this modified Markov process, first, all recurrent classes are singletons, due to the additive condition of this process. Second, all off-diagonal points can never appear in the absorbing state (similar to the Claim 1 in the proof of Proposition 1). Third, any absorbing state must satisfy that there exists an action a such that $Q_i^a = u(a,a)/(1-\delta)$ and $Q_i^{a'} < Q_i^a$, $\forall a' < a$; $Q_i^{a'} \leq Q_i^a$, $\forall a' > a$. The only difference between this and the absorbing state of the original unperturbed Markov process P_0 is that in the modified process, it allows for $Q_i^{a'} = Q_i^a$ for $a' > a$.

Assume that after perturbing player $\bar{i}(s)$ to choose $a' \neq a(s)$, there is a positive probability of transitioning to $\tilde{s}'$, which is an absorbing state of the modified unperturbed process. Let $\underline{\tilde{a}}_{\underline{i}(s)}$ represent the lowest action chosen by $\underline{i}(s)$ in the transition from s to $\tilde{s}'$. Due to the additive action selection rule, either $a(\tilde{s}') = \underline{\tilde{a}}_{\underline{i}(s)}$ or $Q_{\underline{i}(s)}^{\underline{\tilde{a}}_{\underline{i}(s)}}(\tilde{s}') < Q_{\underline{i}(s)}^{\underline{\tilde{a}}_{\underline{i}(s)}}(s)$.

If $\tilde{s}' \in \mathcal{R}$, the proof is complete. Otherwise, if in state $\tilde{s}'$, for some player i, there exists an action $a' > a(\tilde{s}')$ such that $Q_i^{a'} = Q_i^{a(\tilde{s}')}$, we can allow player i to choose a' while player $-i$ continues with their current action. This adjustment transforms $\tilde{s}'$ into one of the absorbing states of the original process, denoted as s', where $a(s') = a(\tilde{s}')$, $Q_{\underline{i}(s)}^{\underline{a}_{\underline{i}(s)}}(s') = Q_{\underline{i}(s)}^{\tilde{\underline{a}}_{\underline{i}(s)}}(\tilde{s}')$, and $\underline{a}_{\underline{i}(s)} = \tilde{\underline{a}}_{\underline{i}(s)}$. □

Suppose in state s', the players switch their labels, i.e., $\bar{i}(s') = \underline{i}(s)$ and $\underline{i}(s') = \bar{i}(s)$. Because $a(s') > a_1$, $Q_{\underline{i}(s)}^{a_1}(s') \leq Q_{\underline{i}(s)}^{a_1}(s)$ (Lemma 6). Hence, $Q_{\underline{i}(s')}^{a_1}(s') \leq Q_{\bar{i}(s')=\underline{i}(s)}^{a_1}(s') \leq Q_{\underline{i}(s)}^{a_1}(s)$. If any of these inequalities is strict, then $s' \prec_g s$ according to condition 1. In the case where the labels are exchanged and $Q_{\underline{i}(s')}^{a_1}(s') = Q_{\bar{i}(s')=\underline{i}(s)}^{a_1}(s') = Q_{\underline{i}(s)}^{a_1}(s)$, the label tie-breaking rule outlined in footnote 11 ensures that if $s' \prec_g s$ due to condition 1 before the label exchange, then $s' \prec_g s$ will still hold due to condition 1 after the label exchange; if $s' \prec_g s$ due to condition 2 before the label exchange, then $s' \prec_g s$ will still hold due to either condition 2 or condition 1 after the label exchange.

If $a(s') > a(s)$, when player $\bar{i}(s)$ is perturbed to choose $g(a(s)) = a'$, then in the next period, player $\bar{i}(s)$ will choose either a' or $a(s)$. Hence, if player $\underline{i}(s)$ never chooses an action below $a(s)$, player $\bar{i}(s)$ will always choose a' or $a(s)$ (Assumption 1.4), which contradicts $a(s') > a(s)$. Then $Q_{\underline{i}(s)}^{a}(s') = Q_{\underline{i}(s)}^{a}(s)$ for all $a < \underline{a}_{\underline{i}(s)}$ and $Q_{\underline{i}(s)}^{\underline{a}_{\underline{i}(s)}}(s') < Q_{\underline{i}(s)}^{\underline{a}_{\underline{i}(s)}}(s)$ (Lemma 6). So, $s' \prec_g s$ (condition 1).

If $a(s') < a(s)$ and player $\underline{i}(s)$ chooses some action lower than $a(s')$, then since the same logic, $s' \prec_g s$ (condition 1). Otherwise, because $a(s') < a(s)$, $s' \prec_g s$ (condition 2 before label exchange).

If $a(s') = a(s)$ and player $\underline{i}(s)$ chooses some action lower than $a(s)$, then since the same logic, $s' \prec_g s$ (condition 1). Otherwise, because in the initial state s, $Q_{\bar{i}(s)}^{a'}(s) < Q_{\bar{i}(s)}^{a'}(s)$ and the instantaneous payoff in each period is $u(a', a'') \geq u(a, a(s)) \geq u(a(s), a(s))$ for all $a'' \geq a(s)$, in state s', $Q_{\bar{i}(s)}^{a'}(s') > Q_{\bar{i}(s)}^{a'}(s)$ (condition 3a). To establish that $s' \prec_g s$, it suffices to demonstrate that, in the extreme case where $Q_{\underline{i}(s')}^{a_1}(s') = Q_{\bar{i}(s')=\underline{i}(s)}^{a_1}(s') = Q_{\underline{i}(s)}^{a_1}(s)$, the label exchange does not occur. This is because $\bar{i}(s)$ selects a' only once due to exploration and never chooses any action lower than $a(s)$ when following the greedy policy. If it were otherwise, given that $\underline{i}(s)$ never selects any action lower than $a(s)$, it would be impossible to converge to a state s' with $a(s') = a(s)$. Consequently, for any action $a'' < a'$, we have $Q_{\bar{i}(s)}^{a''}(s') = Q_{\bar{i}(s)}^{a''}(s)$, and since $Q_{\bar{i}(s)}^{a'}(s')$ is strictly increasing, the tie-breaking rule does not alter the label.

B.6 Proof of Proposition 2

We define $\tilde{a}$ as a valid perturbation with respect to an action $a \neq a_1$ if, for any state s where $a(s) = a$, perturbing player $\bar{i}(s)$ to choose $\tilde{a}$ results in a positive probability of convergence to an absorbing state s' such that $s' \prec_g s$ for some order $\prec_g$. Here, we prove a more general version: when $\delta > 0$, an action $\tilde{a}$ is a valid perturbation w.r.t. action $a \neq a_1$ if and only if $u(\tilde{a}, a) \geq u(a, a)$. The sufficient condition has already been established in the proof of Lemma 5.

Necessity. If this condition is not satisfied, consider a state s where for player $\underline{i}(s)$, $Q^{a'}_{\underline{i}(s)}(s) = \frac{\underline{u}}{1-\delta}$ for all $a' \neq a$ and $Q^{a}_{\underline{i}(s)}(s) = \frac{u(a,a)}{1-\delta}$; for player $\bar{i}(s)$, $\frac{\underline{u}}{1-\delta} < Q^{a'}_{\bar{i}(s)}(s) < Q^{a}_{\bar{i}(s)}(s) = \frac{u(a,a)}{1-\delta}$ for all $a' \neq a$, and $Q^{\tilde{a}}_{\bar{i}(s)}(s) > \frac{(1-\delta)u(\tilde{a},a)+\delta u(a,a)}{1-\delta}$. Since $u(\tilde{a}, a) < u(a, a)$, after a perturbation, player $\bar{i}(s)$ will revert to selecting a. For player $\underline{i}(s)$, after updating, the Q-value $Q^{a}_{\underline{i}(s)}$ will lie between $u(a, \tilde{a}) + \delta Q^{a}_{\underline{i}(s)}(s)$ and $Q^{a}_{\underline{i}(s)}(s)$, both of which are strictly greater than $\frac{\underline{u}}{1-\delta}$, the Q-values of other actions. Therefore, player $\underline{i}(s)$ will continue to select a.

In the next absorbing state s', the only change will be that $Q^{\tilde{a}}_{\bar{i}(s)}(s') < Q^{\tilde{a}}_{\bar{i}(s)}(s)$, while all other Q-values remain unchanged. Since $Q^{a_1}_{\underline{i}(s)}(s') = \frac{\underline{u}}{1-\delta}$ and (including the case that $\tilde{a} = a_1$) $Q^{a_1}_{\bar{i}(s)}(s') \geq \min\left\{Q^{a_1}_{\bar{i}(s)}(s), u(a_1, a) + \delta\frac{u(a,a)}{1-\delta}\right\} > \frac{\underline{u}}{1-\delta}$, $\bar{i}(s') = \bar{i}(s)$ and $\underline{i}(s') = \underline{i}(s)$. Thus, there does not exist an order $\prec_g$ s.t. $s' \prec_g s$: if $g(a) = a'$, then $s \prec_g s'$; otherwise, s and s' cannot be compared.

C Simulation Design

C.1 Economic Environment

In the simulation, we focus on a Bertrand competition model. Consider two firms, $i \in I = \{1, 2\}$, with equal marginal costs normalized to 0. These firms sell homogeneous goods and compete by setting prices, which are determined using Q-learning. The value of the good to the consumer is normalized to 1. The price set by firm i is denoted as $p_i \in A$. We assume that consumers will buy from the firm offering the lowest price. In the event of a tie, demand is split equally between the firms. The demand faced by firm i is parameterized as follows:

$$q_i(p_i, p_j) = \begin{cases} 1 & \text{if } p_i < p_j \text{ and } p_i \leq 1; \\ \frac{1}{2} & \text{if } p_i = p_j \text{ and } p_i \leq 1; \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Hence, the immediate payoff of the firm i is $u_i(p_i, p_j) = p_i q_i(p_i, q_j)$.

In the baseline setting, we set the number of actions, $K = 10$, with the action space $A = \{0.1, 0.2, \ldots, 0.9, 1.0\}$. Here, 0.1 represents the unique strict NE, while 0.2 is another NE.

C.2 Initialization

As for the initial matrix $\mathbf{Q}_0$, our baseline choice is to set the Q-values at $t = 0$ at the discounted payoff that would accrue to player i if opponents randomized uniformly (like Calvano et al. 2020).[21] For any $i \in I$ and $a \in A$,

$$Q_i^a(s_0) = \frac{\sum_{a' \in A} u(a, a')}{(1-\delta)|A|}. \tag{20}$$

C.3 Policy

Regarding the exploration mode, we use the ε-greedy model with a time-declining exploration rate in the baseline.[22] Specifically, we set $\varepsilon_t = e^{-\beta t}$, where $\beta > 0$ is *decay rate*. In period t, Q-learning selects the optimal action $a^* \in \arg\max_{a' \in A} Q_i^{a'}(s_t)$ with probability $1 - \varepsilon_t$, and with probability ε_t, it randomly chooses an action $a \in A$.

In the baseline simulation setup, to establish reasonable values for β, it is useful to relate β to the expected number of times a cell would be visited purely through random exploration over an infinite time horizon, denoted by ν (like Calvano et al. 2020). In the single-state without memory case, we have

$$\nu = -\frac{1}{K\left(1 - e^{-\beta}\right)}. \tag{21}$$

For our baseline scenario, we set $\beta = 1 \times 10^{-4}$, which corresponds to $\nu \approx 1000$ when $K = 10$.

[21] Another commonly used initialization methods include zero initialization, where all Q-values are set to zero; random initialization, where Q-values are initialized randomly, typically using a uniform or normal distribution; and optimistic initialization, where Q-values are initialized to high positive values. In Asker et al. (2022, 2023), they use optimistic initialization with a uniform distribution, specifically setting $Q_i^a(s_0) \sim_{i.i.d.} U(1, 2)$.

[22] Another exploration policy common used is Boltzmann exploration. In Boltzmann exploration, the probability of selecting an action a at time t is determined by the Boltzmann (softmax) distribution:

$$\Pr_i(a) = \frac{e^{Q_i^a(s_t)/\tau_t}}{\sum_{a' \in A} e^{Q_i^a(s_t))/\tau_t}}, \ \forall i \in I \text{ and } \forall a \in A.$$

where $\tau_t > 0$ is the temperature parameter at time t (that can decay over time).

C.4 Simulation with Numerous Periods

In Subsection 4.1, we conduct simulations under a slight and constant exploration rate. To manage the extensive number of periods, $\frac{T}{\varepsilon}$, within each unit of periods, $\frac{1}{\varepsilon}$, we employ a binomial distribution, $B\left(\frac{1}{\varepsilon}, \varepsilon\right)$, to generate the sequence of periods during which each player will explore randomly. Once the instances of exploration are determined, in the intervals between consecutive explorations, if a player repeatedly selects the same action a according to the greedy policy and $\frac{u(a,a)}{1-\delta}$ is greater than the second-highest Q-value, we skip this interval of length τ. We update the Q-value of a as follows:

$$Q_i^a(s_{t+\tau}) = (1-\alpha(1-\delta))^\tau \, Q_i^a(s_t) + [1-(1-\alpha(1-\delta))^\tau] \, \frac{u(a,a)}{1-\delta}. \tag{22}$$

This approach allows us to efficiently simulate the behavior over long periods while maintaining the accuracy of the learning process.

D Figures

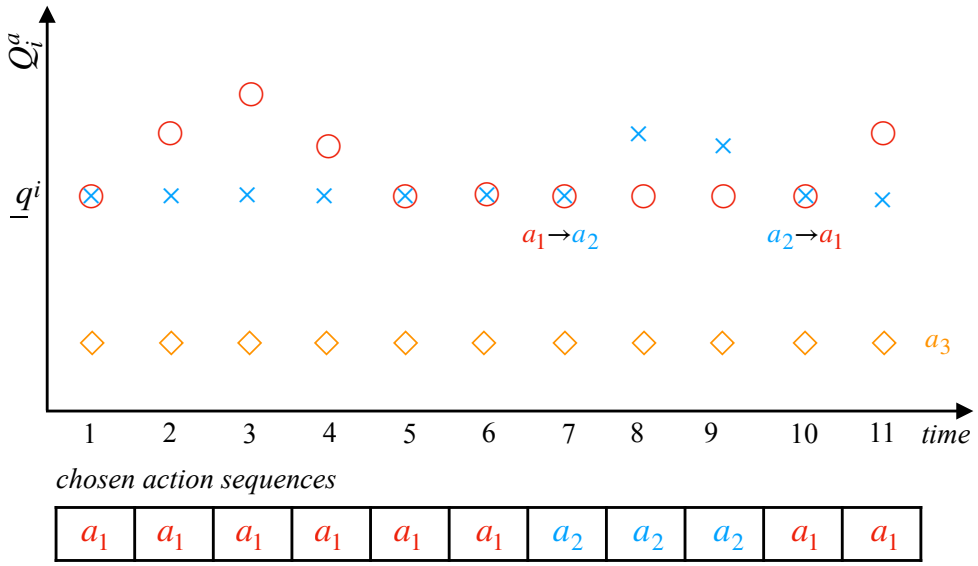


Figure 1: Q-value and Action Transition

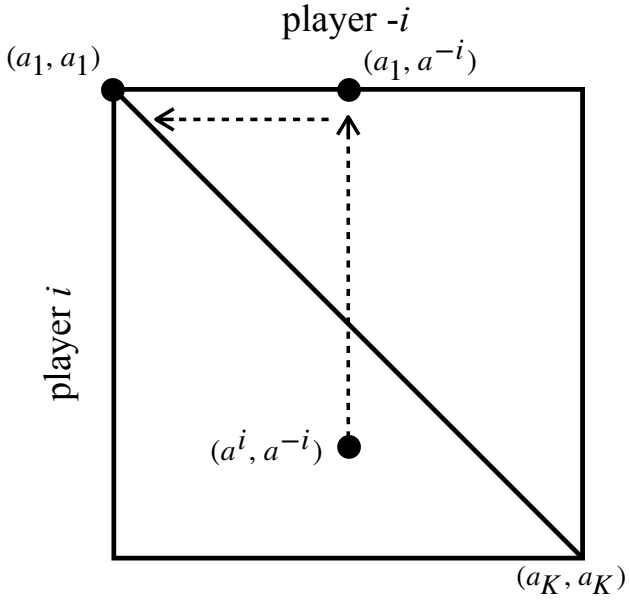
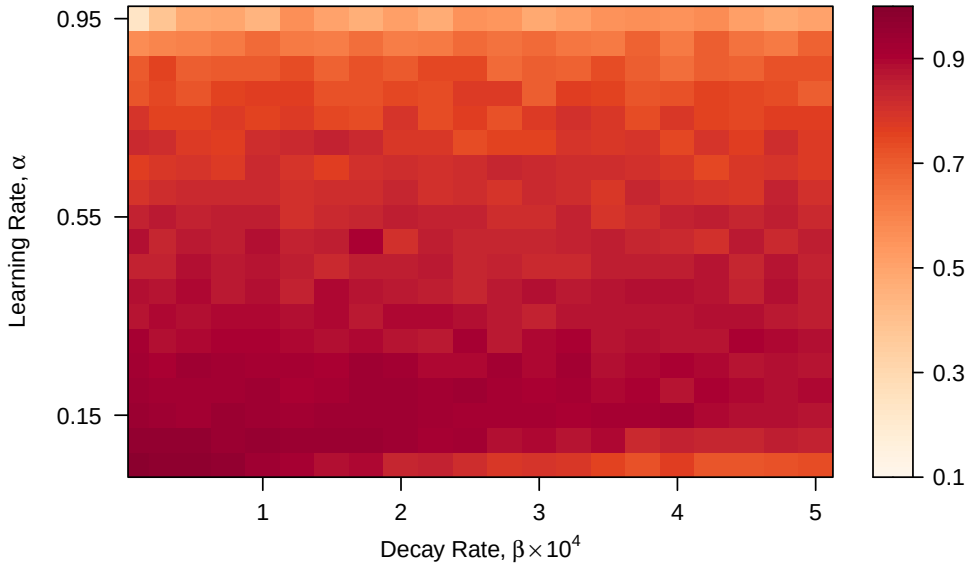
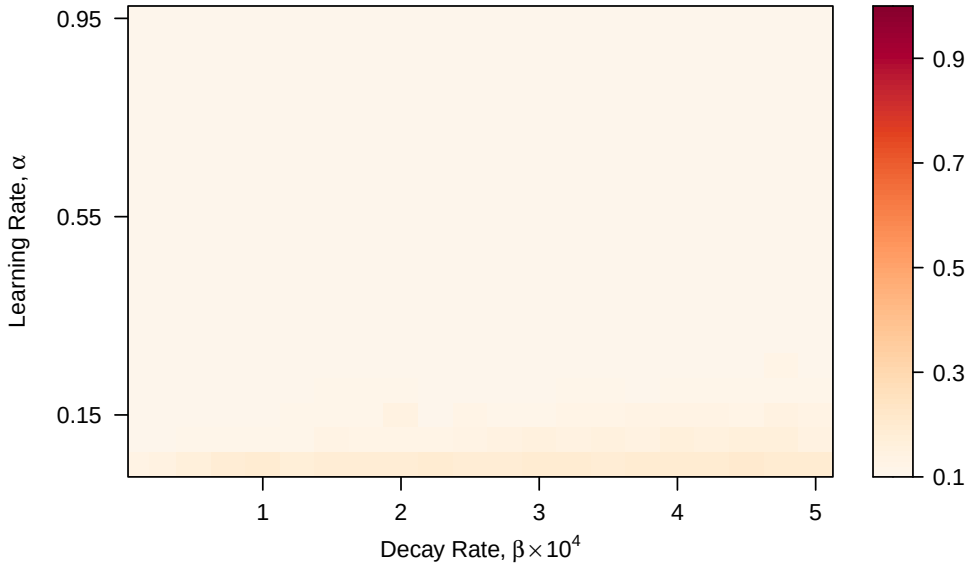


Figure 2: Illustration of the Proof of Proposition 1

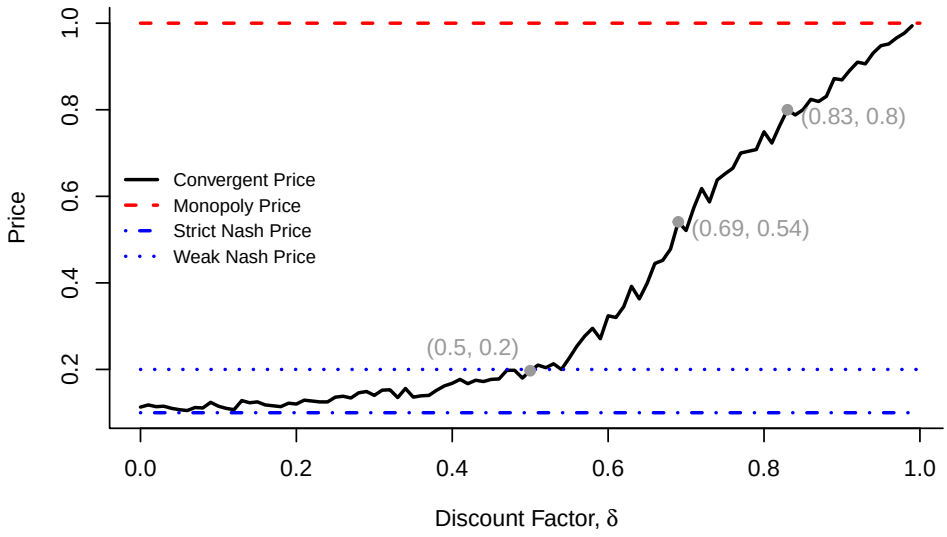


(a) $\delta = 0.95$

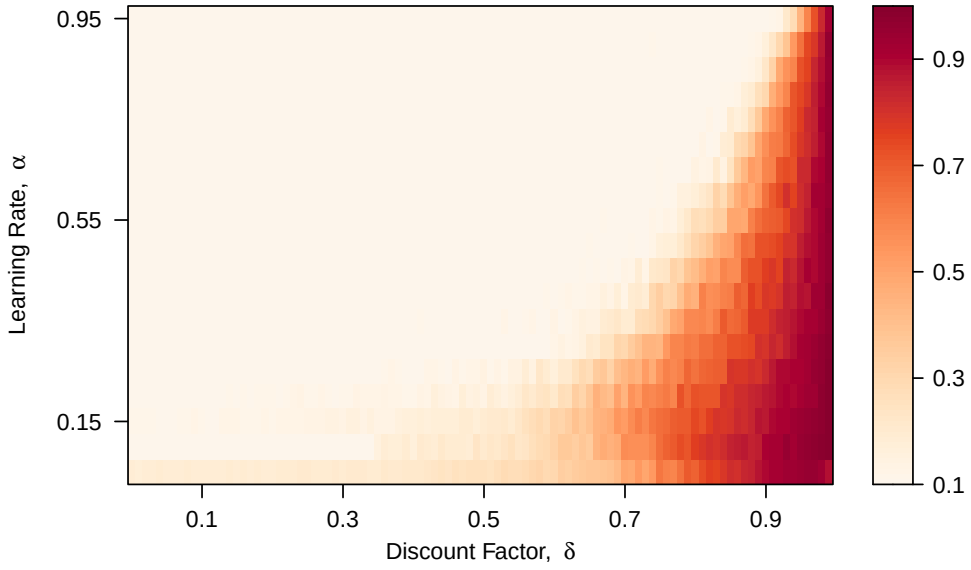


(b) $\delta = 0$

Figure 3: Average Convergent Price for a Grid of Values of α and β (Decay)

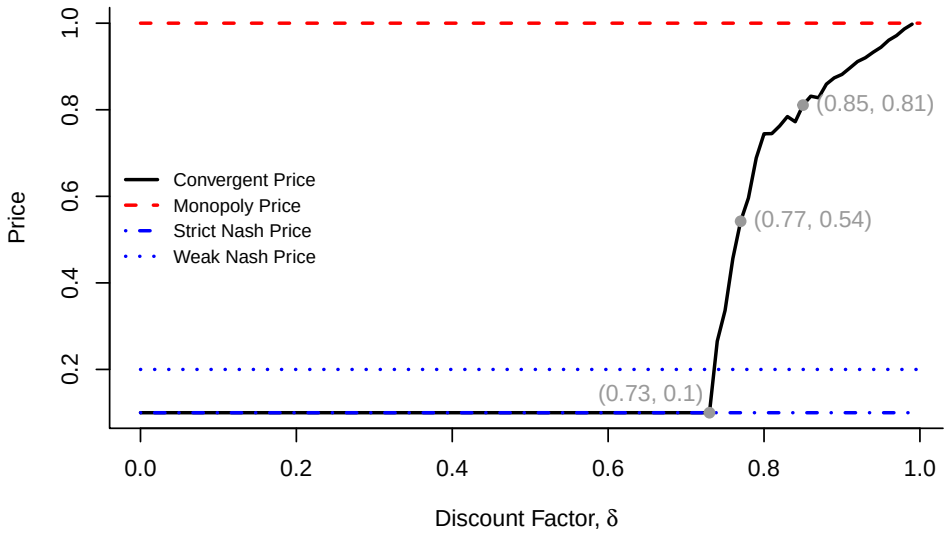


(a) $\alpha = 0.15$

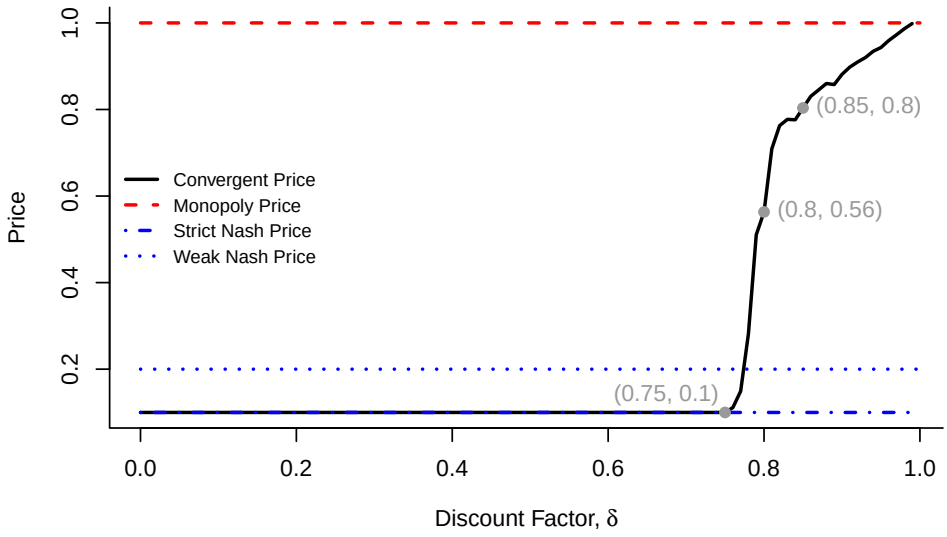


(b) Robust Check, $0.05 \leq \alpha \leq 0.95$

Figure 4: Average Convergent Price as a Function of δ (Decay)



(a) $\varepsilon = 10^{-6}$, $T = 10^5$



(b) $\varepsilon = 10^{-6}$, $T = 10^6$

Figure 5: Average Convergent Price as a Function of δ (Constant)

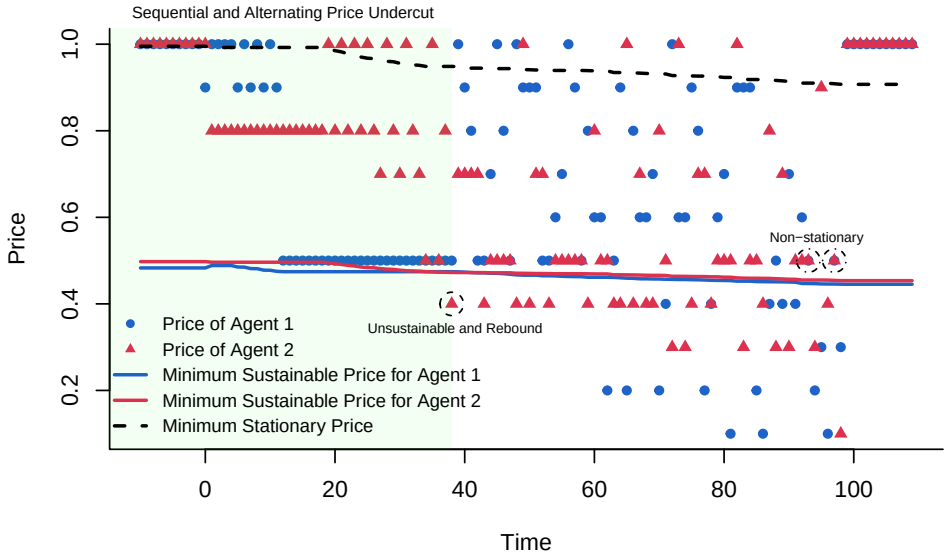


Figure 6: Illustration of Price Undercut and Rebound (Decay)

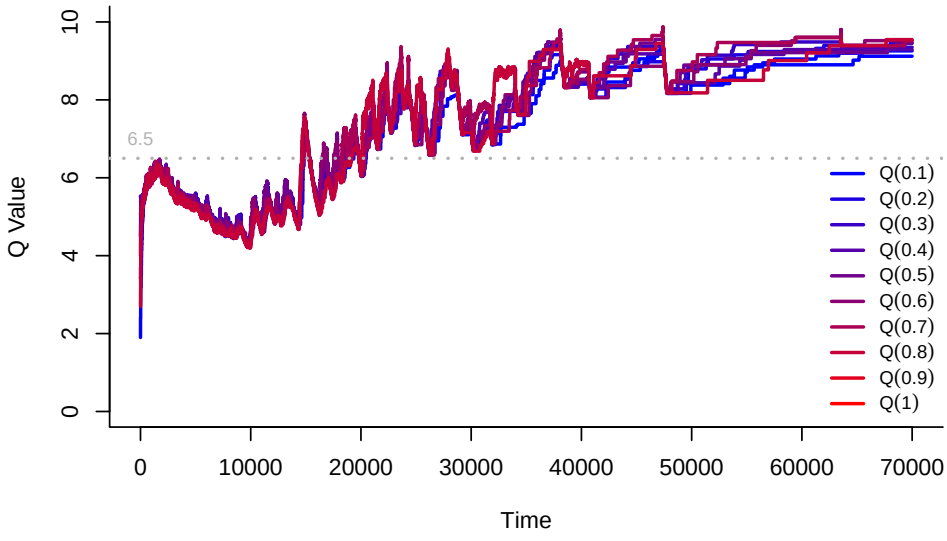
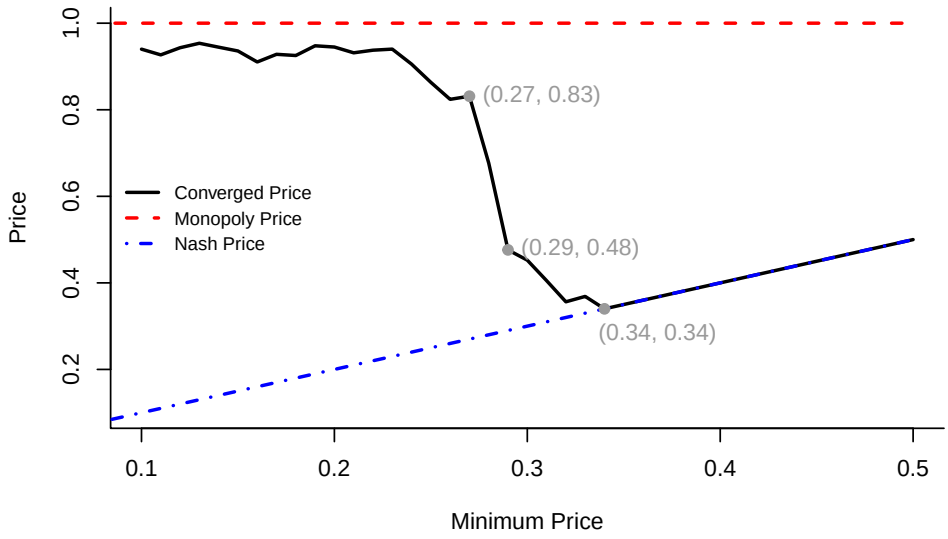
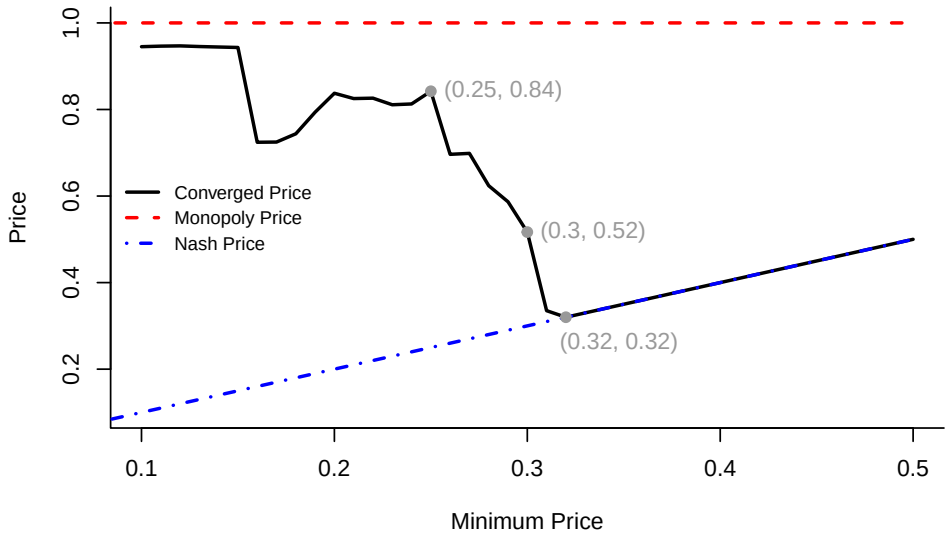


Figure 7: Illustration of Q-value Evolution (Decay)



(a) Decay



(b) Constant, $\varepsilon = 10^{-6}$, $T = 10^5$

Figure 8: Average Convergent Price as a Function of Minimum Price

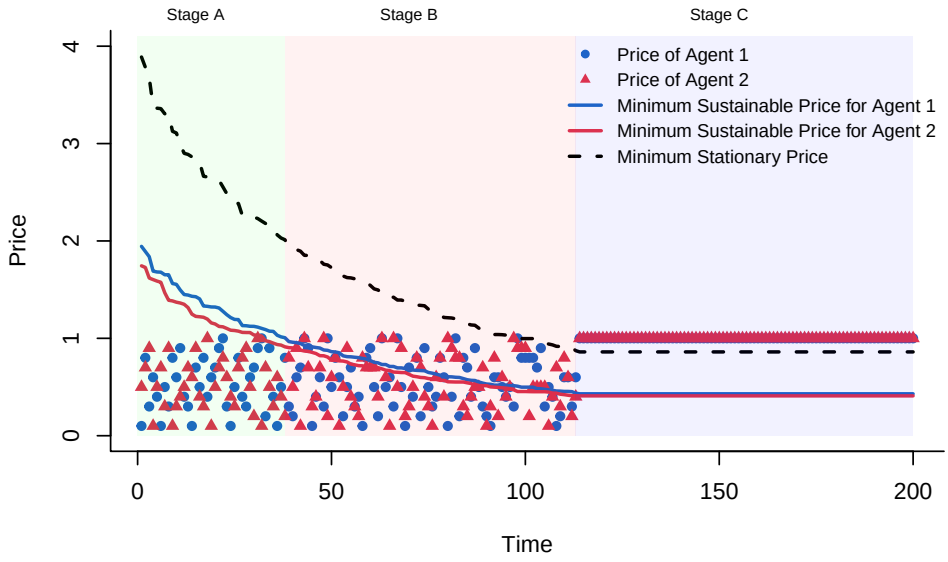


Figure 9: Illustration of Price Undercut and Rebound of Optimal Initialization

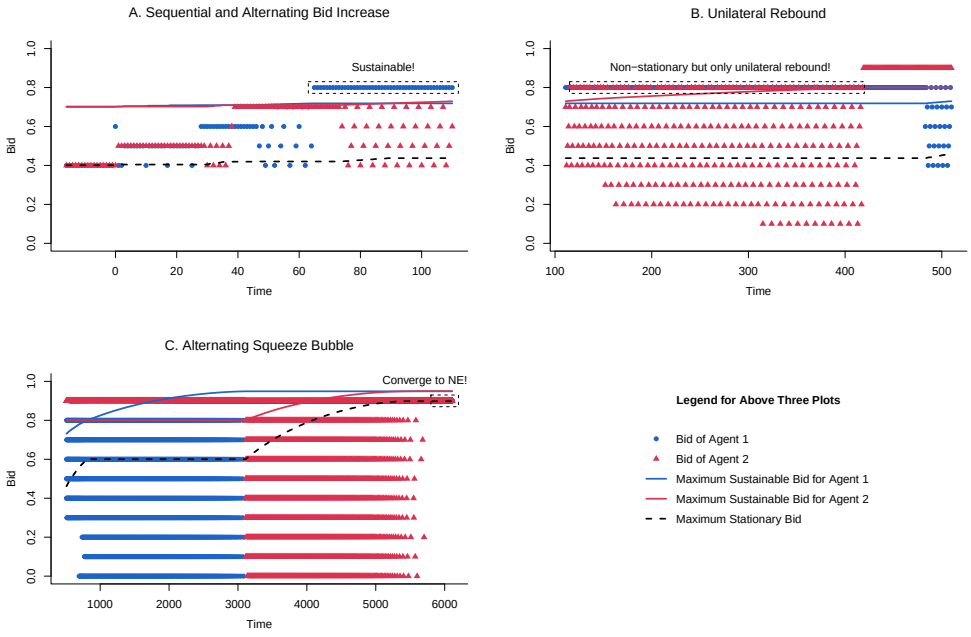
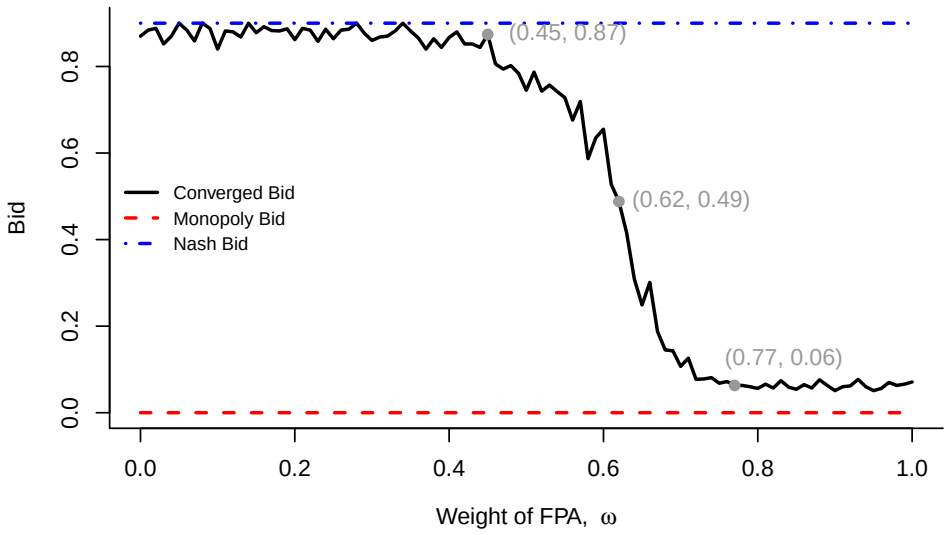
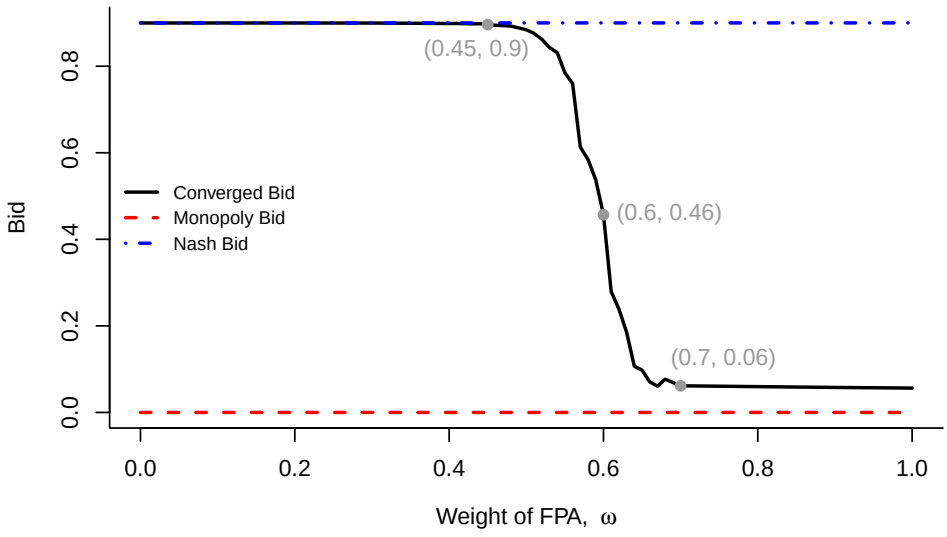


Figure 10: Illustration of Bid Increase and Rebound of Second Price Auction

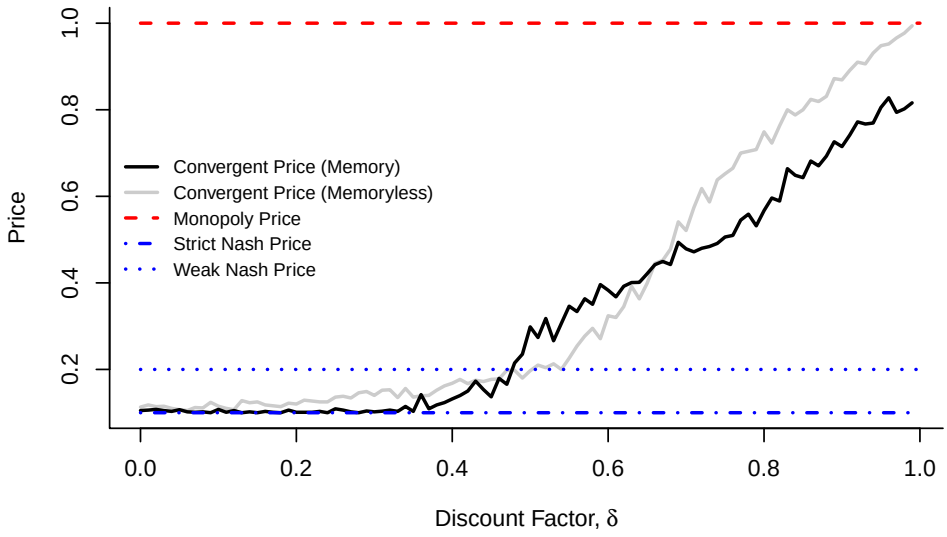


(a) Decay

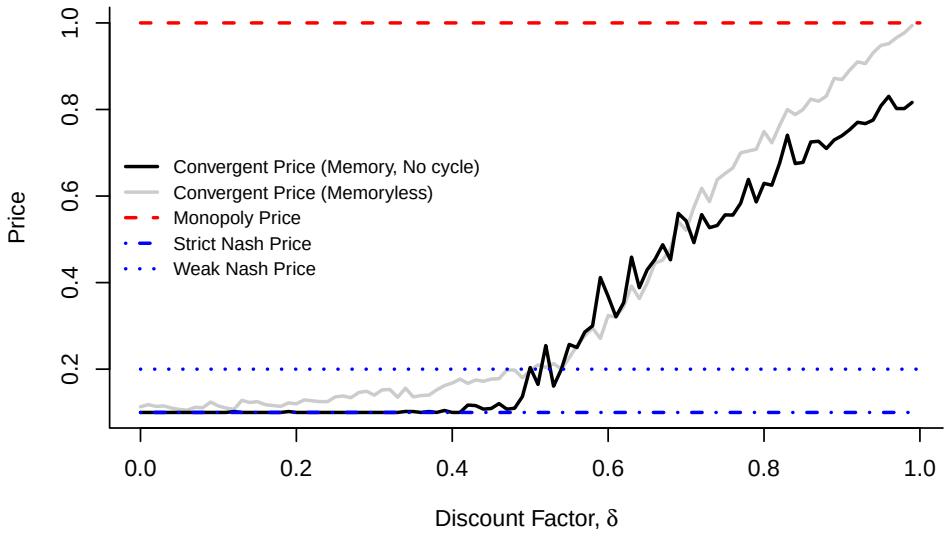


(b) Constant, $\varepsilon = 10^{-6}$, $T = 10^5$

Figure 11: Average Convergent Bid as a Function of the Weight of FPA



(a) w Cycle



(b) w/o Cycle

Figure 12: Effect of Memory on the Convergent Price as a Function of the Discount Factor

Note: When a one-period memory is introduced into Q-learning, the Q-vector, which initially has 10 cells, is extended to a Q-matrix with 1000 cells. Consequently, this necessitates more exploration compared to the memoryless case to ensure equivalent random exploration of each cell. On the other hand, given the computational limitations, the decay rate in the simulation of the memory case is set to be one-tenth of the value used in the memoryless case, i.e., $\beta = 1 \times 10^{-5}$. All other parameters remain the same. In (a), we present the average convergent price across all outcomes, whereas in (b), we focus on the average convergent price specifically among the outcomes that converge to a single price without cycling.

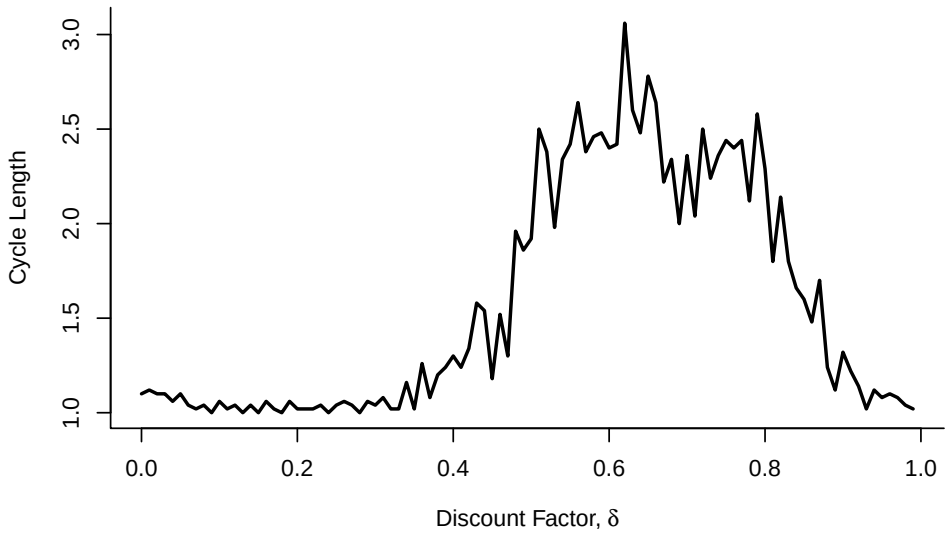


Figure 13: Cycle Length of Convergent Outcome in the Memory Case (Decay)