# Algorithmic Collusion under Observed Demand Shocks

Zexin Ye *

February 22, 2025

**Abstract**

When the current demand shock is observable, with a high discount factor, Q-learning agents predominantly learn to implement symmetric rigid pricing, i.e., they charge constant prices across demand states. Under this pricing pattern, supra-competitive profits can still be obtained and are sustained through collusive strategies that effectively punish deviations. This shows that Q-learning agents can successfully overcome the stronger incentives to deviate during the positive demand shocks, and consequently algorithmic collusion persists under observed demand shocks. In contrast, with a medium discount factor, Q-learning agents learn that maintaining high prices during the positive demand shocks is not incentive compatible and instead proactively charge lower prices to decrease the temptation for deviating, while maintaining relatively high prices during the negative demand shocks. As a result, the countercyclical pricing pattern becomes predominant, aligning with the theoretical prediction of Rotemberg and Saloner (1986). These findings highlight how Q-learning algorithms can both adapt pricing strategies and develop tacit collusion in response to complex market conditions.

**Keywords:** algorithmic collusion, observed demand shocks, price rigidity, countercyclical pricing

**JEL Codes:** D21, D43, D83, L13

*ye.754@osu.edu, Department of Economics, The Ohio State University. I would like to thank Huanxing Yang for his valuable advice. I am also grateful to Yonghong An, Yaron Azrieli, Paul J. Healy and Lixin Ye for their insightful comments. Additionally, I appreciate the worthwhile feedback from the Micro reading group at Ohio State. All errors remain my own.

# 1 Introduction

In recent years, concerns have grown among researchers and policymakers regarding the potential impact of AI-powered pricing algorithms on market competition and consumer welfare. A prominent example is the recent U.S. housing rental market, where widespread adoption of algorithmic pricing has inflated rental prices, affecting millions of households and adding a total of $3.8 billion to annual rental expenditures.[1] In response to these concerns, broader legislative efforts are underway through Senate Bill S.3686 (Preventing Algorithmic Collusion Act), which aims to prohibit the use of nonpublic competitor data and enhance algorithmic transparency.[2]

On the research side, many studies have shown that AI-powered pricing algorithms can autonomously learn collusive strategies and charge supra-competitive prices, even without explicit programming, direct communication, or access to nonpublic data of rivals.[3] While existing simulation studies have provided valuable insights into algorithmic collusion, they typically assume an unchanging economic environment. Further exploration is needed to understand whether algorithmic collusion can persist in more complex market conditions, e.g., markets with demand shocks. Calvano et al. (2021) address this complexity by examining settings where demand shocks are unobserved and monitoring is imperfect.[4]

However, only considering unobserved demand shocks is not enough. In business practice, precise prediction of market demand is crucial for guiding critical business decisions, such as production planning, inventory management, and pricing strategies. Firms increasingly rely on advanced algorithms and machine learning techniques to generate accurate

[1] This has prompted both legal action against real estate companies and legislative responses. See https://www.justice.gov/archives/opa/media/1364976/dl?inline and https://www.congress.gov/bill/118th-congress/senate-bill/3692.

[2] See https://www.congress.gov/bill/118th-congress/senate-bill/3686.

[3] Assad et al. (2021) provide a review of algorithmic collusion, discussing recent developments in the economic literature and their policy implications. Recent research has also explored potential remedies to mitigate algorithmic collusion (Beneke and Mackenrodt, 2021; Brero et al., 2022).

[4] Their findings demonstrate that imperfect monitoring does not necessarily prevent algorithmic collusion. Calvano et al. (2020) demonstrate algorithmic collusion is robust under unobserved demand shocks with perfect monitoring.

demand forecasts.[5] In light of this widespread practice, the impact of demand forecasting on algorithmic collusion remains unexplored, and its potential outcomes are uncertain. On the one hand, predicting future demand shocks can reduce uncertainty and thus facilitate collusive coordination. On the other hand, access to more information may hinder such coordination. Specifically, if algorithms can observe demand shocks, they may be tempted to deviate from collusive pricing during periods of the positive demand shocks.

To partially address this research gap, I assume algorithms can perfectly predict the current demand state. Thus, I am able to adopt the framework of Rotemberg and Saloner (1986) to investigate algorithmic collusion under observed demand shocks. In this model, agents participate in an infinitely repeated Bertrand competition with a homogeneous good, where an i.i.d. demand shock occurs in each period. In each period, agents first observe the current realized demand state (perfect prediction) and then set prices simultaneously. Moreover, agents can observe their rivals' past prices, making this a perfect monitoring setting. In the baseline model, the demand shock consists of two levels, positive and negative, each occurring with equal probability.

The agents are played by identical Q-learning algorithms, which are commonly used in previous studies on algorithmic pricing. Q-learning is a fundamental algorithm in reinforcement learning, known for its simplicity and model-free nature.[6] I modify the Q-learning algorithm to incorporate the current realized demand state, ensuring that the algorithms truly "observe" the demand shocks.[7]

My main findings are twofold. First, when agents are sufficiently patient (i.e., the discount factor is high), Q-learning algorithms strongly prefer charging constant prices across demand states, resulting in price rigidity. This is contrary to procyclical pricing predicted

[5]For example, in supply chain management (Seyedan and Mafakheri, 2020) and electricity markets (Raza and Khosravi, 2015).

[6]Reinforcement learning (RL) has emerged as a foundational technique of modern artificial intelligence, with industry leaders like OpenAI and ByteDance actively deploying RL-based frameworks to develop advanced AI models. Sutton (2018) provides a comprehensive introduction to RL.

[7]See details in Section 3.

by Rotemberg and Saloner (1986).[8] Under this pricing pattern, the agents autonomously learn collusive strategies and consistently achieve supra-competitive profit without explicit programming or communication. Deviation tests reveal that the expected losses exceed the expected gains for any possible price undercut from any demand state, and the probability of profitable deviations consistently remains low. This suggests that Q-learning agents can successfully maintain collusion even when facing unbalanced deviating incentives. Thus, algorithmic collusion persists under observed demand shocks. Moreover, the price dynamics on the deviation path exhibit distinct patterns. After the initial deviation, the price drops quickly and persists at low levels (frequently at the competitive price level) for sufficiently long periods, until reaching "restart points" that trigger a swift return to the pre-deviation price. This exhibits a discontinuous switching pattern between punishment and cooperation phases, contrasting with Calvano et al. (2020) where the price on the deviation path gradually returns to the pre-deviation level.

Second, when the discount factor is at a medium level, countercyclical pricing becomes predominant instead of rigid pricing, consistent with the prediction of Rotemberg and Saloner (1986). Specifically, Q-learning agents correctly recognize that high prices are not sustainable during the positive demand shocks and respond by proactively lowering prices, while maintaining relatively high prices during the negative demand shocks. A distinctive feature of this pricing pattern is that the average during the positive demand shocks are particularly low (approaching the competitive price level), resulting in almost no profitability. Nevertheless, the overall profits remain significantly higher than under rigid pricing, rationalizing the predominant choice of countercyclical pricing.

The robustness checks demonstrate that algorithmic collusion occurs across an extensive range of parameters, rather than relying on artificial parameter selections. Similarly, the emergence of specific pricing patterns remains stable across different parameter values. Furthermore, the analysis reveals a necessary condition for countercyclical pricing:

[8]The reason lies in whether past realized of demand shocks are included in state variables. For more details, see the Discussion in Section 4.2.

both agents must observe the demand shocks. In an asymmetric setting where only one agent observes the demand shocks, countercyclical pricing cannot be sustained. This occurs because the informed agent's low prices during the positive demand shocks would be misinterpreted as deviations by the uninformed agent, triggering price wars across demand states and ultimately leading to uniformly low prices across demand states.

This study enriches the research on algorithmic collusion, demonstrating that Q-learning algorithms can adapt pricing strategies to different market conditions and maintain tacit collusion under observed demand shocks. To the best of my knowledge, this is the first study to incorporate observed demand shocks in the context of algorithmic collusion. My findings also provide experimental support for the pooling scheme in algorithmic pricing.[9] These findings have important policy implications. The observation of lower prices during booms than busts complicates the identification of algorithmic collusion and price coordination. This complexity necessitates further research and underscores the importance of economic theory in understanding these phenomena and guiding policy interventions in algorithmic pricing.

The rest of the paper is organized as follows. Section 2 provides a review of the literature. Section 3 introduces the economics environment for simulation and then explains in detail the modified Q-learning algorithms that can observe demand shocks. Section 4 describes how to derive the long-run price cycles, based on which I analyze the pricing patterns, evaluate the performance and conduct the deviation tests. Section 5 conducts a number of robustness checks. Section 6 concludes.

[9] In collusion literature, there are two main collusive pricing schemes: sorting and pooling (rigid-pricing). For theoretical analysis on collusion and price rigidity, Athey et al. (2004) study a setting in which each agent experiences a privately i.i.d. cost shock each period. Hanazono and Yang (2007) assume that each agent observes an independent private signal about the underlying demand state each period and find that collusion can be maintained through price rigidity when the signals have low accuracy.

# 2 Literature Review

In this section, I first review recent studies on algorithmic collusion, then examine research on AI-powered algorithms across broader economics topics.

The research on algorithmic collusion gains significant momentum with the pioneering work of Calvano et al. (2020), who study agents driven by Q-learning algorithms in a general Bertrand oligopoly model incorporating both vertical and horizontal differentiation. Their simulations reveal that agents autonomously learn to set supra-competitive prices. Moreover, these supra-competitive prices are maintained by collusive strategies that effectively deter deviations. In a dynamic sequential pricing framework adapted from Maskin and Tirole (1988), where agents set prices in turns, Klein (2021) demonstrates that Q-learning algorithms can still achieve stable price collusion.

However, the two studies mentioned above use simplified economic environments that do not consider how AI-powered agents behave under demand shocks. This limitation may restrict the implications of their findings on algorithmic collusion in more complex, real-world settings. Traditional Industrial Organization (IO) theory offers two primary frameworks for incorporating demand shocks into models of (tacit) collusion, which are discussed below.

In the framework developed by Green and Porter (1984), agents cannot perfectly monitor rivals' behavior and face an i.i.d. unobserved demand shock each period. Thus, agents cannot clearly infer whether low market prices result from negative demand shocks or rivals' output expansion (Cournot competition). The collusive strategies they characterize involve temporary price wars on the equilibrium path triggered by unexpectedly low market prices. Such price wars serve as a punishment mechanism to discourage firms from deviating. Building on this framework, Calvano et al. (2021) show that Q-learning algorithms can still maintain tacit collusion, with the collusive strategies learned being remarkably similar to those in Green and Porter (1984). This demonstrates that autonomous algorithmic collusion can be achieved even under imperfect monitoring, underscoring its resilience to uncertain market conditions.

In the other framework, Rotemberg and Saloner (1986) treat the i.i.d. demand shock in each period as observed, and agents can perfectly monitor their rivals' behavior. They show that when agents are sufficiently patient (i.e., the discount factor is large), the best collusive outcomes can be sustained, with agents charging monopoly prices at each demand state, resulting in procyclical pricing (i.e., prices move in the same direction of the economic cycle). When the discount factor is not sufficiently large, there may exist downward pricing distortion because deviating from collusive outcomes during booms is so profitable that agents must lower prices to reduce the temptation to deviate. Thus, unlike Green and Porter (1984), Rotemberg and Saloner (1986) predict that price wars occur in booms instead of busts. In some cases, prices during booms can be even lower than during busts, forming countercyclical pricing (i.e., prices move in the opposite direction of the economic cycle).

There have been numerous theoretical studies following the seminal work of Rotemberg and Saloner (1986). In the model of Rotemberg and Saloner (1986), it is assumed that agents are able to directly observe the current demand state, implying that their prediction of the current demand state is perfect. In contrast, recent studies by Miklós-Thal and Tucker (2019) and O'Connor and Wilson (2021) theoretically examine how improved demand prediction capabilities (i.e., improved precision of signals about the current demand state) affect collusive outcomes. Specifically, Miklós-Thal and Tucker (2019) show that agents have an increased incentive to undercut their rivals during periods of high demand that are more accurately predicted. With respect to demand shocks, the i.i.d. assumption in Rotemberg and Saloner (1986) implies that agents' expectations of future demand are independent of current demand. Kandori (1991) demonstrates that the countercyclical pricing pattern remains in the presence of serially correlated Markov demand shocks. Similarly, Haltiwanger and Harrington Jr (1991) study pricing behavior throughout the business cycle and show that, for a given level of demand, prices are lower when market demand is decreasing compared to when market demand is increasing, thus exhibiting countercyclical pricing. My paper builds on the framework of Rotemberg and Saloner (1986). Although the model abstracts

from some market complexities, it provides a good starting point for studying algorithmic collusion under observed demand shocks.

In a broader context, this study contributes to the rapidly growing literature on the interaction between economics research and artificial intelligence. To name a few, Fish et al. (2024) show that agents powered by Large Language Models (LLMs) can maintain autonomous collusion without explicit instructions. Assad et al. (2024) empirically show that the widespread introduction of algorithmic pricing in Germany's retail gasoline market significantly raises profit margins (retail prices over wholesale prices), thereby softening competition. Ballestero (2021) demonstrates algorithmic collusion under the setting of sequential pricing with stochastic costs. Research on algorithmic fairness has also emerged as an active topic (Jabbari et al., 2017; Rambachan et al., 2020; Cowgill and Tucker, 2020; Liang et al., 2021). In the context of platform design, Johnson et al. (2023) develop demand-steering rules, which are algorithmic techniques used by platforms to guide consumer demand towards certain products or services. Through simulations based on reinforcement learning algorithms, they demonstrate that these rules can improve consumer welfare and increase platform revenue. Dolgopolov (2024) characterize the outcomes in a prisoner's dilemma, where the competing agents are played by the reinforcement learning algorithms, showing that cooperation is possible when the algorithms have a high learning rate (i.e., they quickly adapt to new information) and do not condition on history (i.e., they do not use past interactions to inform their decisions). Xu and Zhao (2024) explore the mechanism behind algorithmic collusion in a general class of symmetric games. Finally, in the field of auctions, Banchio and Skrzypacz (2022) find that with limited information, Q-learning algorithms learn to bid low in first-price auctions but not in second-price auctions. Kolumbus and Nisan (2022) study the behavior of regret-minimizing algorithms in different auction settings.

# 3 Experimental Design

## 3.1 Economic Environment

Two agents engage in an infinitely repeated Bertrand pricing competition, each producing a homogeneous product under linear demand.[10] In every period $t$, a random common demand shock $\theta_t$ occurs, shifting the market demand curve parallelly. This shock is i.i.d. and follows a uniform distribution over $[\underline{\theta}, \overline{\theta}]$; Both agents first observe the realized $\theta_t$, and then set their prices simultaneously. Let $\mathbf{p}_t = (p_{1t}, p_{2t})$ be the complete price profile and $p_{-it}$ be the rival price. The demand for agent $i$ at period $t$ is given by

$$D_{it}(p_{it}, p_{-it}, \theta_t) = \begin{cases} b + \theta_t - p_{it} & \text{if } p_{it} < p_{-it} \\ \dfrac{b + \theta_t - p_{it}}{2} & \text{if } p_{it} = p_{-it} \\ 0 & \text{if } p_{it} > p_{-it} \end{cases}$$

where $b$ is the coefficient of the market demand. Correspondingly, the period payoff for agent $i$ at period $t$ is $\pi_{it} = (p_{it} - c_i) D_{it}(p_{it}, p_{-it}, \theta_t)$, where $c_i$ is the constant marginal cost.

**Dynamic Problem** In the infinitely repeated Bertrand game, agent $i$'s problem is to maximize the expected present value of the payoff stream $\sum_{t=0}^{\infty} \delta^t \pi_{it}$ with discount factor $\delta$. Agent $i$'s problem can be modelled as a Markov decision process, in which the agent chooses price $p_{it}$ based on state $s_t$. The value function for agent $i$ is

$$V_i(s) = \max_{p_i \in A} \left\{ \pi_i + \delta \mathbb{E}\left[ V_i(s') \mid s, p_i \right] \right\} \tag{1}$$

where $s'$ is the next state and $A$ is the action (price) space.

[10] This simplified setting builds on the framework of Rotemberg and Saloner (1986), enabling a focused analysis of countercyclical pricing. For reference, Calvano et al. (2020) use a logit demand function that captures both horizontal and vertical differentiation in a generalized duopoly setting, while Calvano et al. (2021) consider a Cournot oligopoly model with stochastic demand.

**Action** Q-learning requires a finite action space. Therefore, I discretize the action space into $m$ equally spaced points within $[\underline{p}^C, \overline{p}^M]$, where $\underline{p}^C$ is the Bertrand equilibrium price under $\underline{\theta}$ and $\overline{p}^M$ is the monopoly price under $\overline{\theta}$.

**State** The state $s_t$ should contain two key components. First, it should include the current demand state $\theta_t$, allowing agents to observe and respond to market fluctuations. Second, under perfect monitoring, it should include past prices to enable agents to detect and punish deviations so that tacit collusion can be sustained.

To prevent the state space from growing indefinitely, the model employs a bounded memory of $K$ periods. This means agents only remember the prices from the last $K$ periods.[11] The resulting state $s_t$ is given by

$$s_t = \{\mathbf{p}_{t-1}, \ldots, \mathbf{p}_{t-k}, \theta_t\}$$

where the demand state $\theta_t$ is realized at the beginning of period $t$.

Notably, past realizations of demand shocks are excluded from $s_t$.[12] This specification is based on three theoretical justifications. First, given that the demand shocks are i.i.d., past shocks provide no predictive value for the current demand state $\theta_t$. Second, past demand shocks neither influence the current period payoff nor help detect deviations in rival's pricing. Third, excluding past shocks reduces the state space dimensionality, thereby enhancing computational efficiency and facilitating agents' learning.

[11] This restriction is justified by both theoretical and simulation results. Barlo et al. (2009) and Barlo et al. (2016) have proved the folk theorems with bounded memory in infinite and finite action spaces, respectively. Calvano et al. (2020) demonstrate the collusive outcomes achieved by Q-learning algorithms with bounded memory.

[12] Including past realizations of demand shocks in $s_t$ does not affect algorithmic collusion but changes the predominant pricing pattern when $\delta$ is high (see the Discussion in Section 4.2).

## 3.2 Q-Learning Algorithms

The value function can be expressed in terms of a Q-function that represents the expected discounted value (Q-value) associated with selecting price $p$ in state $s$, as follows:

$$Q_i(s,p) = \pi_i + \delta \mathbb{E}\left[\max_{p' \in A} Q_i(s',p') \mid s,p\right] \tag{2}$$

where the first term on the right-hand side is the period payoff and the second term is the expected continuation value ($s'$ denotes the next-period state).[13] Since $S$ and $A$ are finite, the Q-function can be represented as an $|S| \times |A|$ matrix, where each entry in the Q-matrix $\mathbf{Q}_i$ stores the corresponding Q-value $Q_i(s,p)$.

However, the Q-matrix $\mathbf{Q}_i$ cannot be solved directly, since the period payoff $\pi_i(p_i, p_{-i}, \theta)$ depends on the rival's pricing decision, and the transition function $F_i(s' \mid s,p)$ remains unknown to agent $i$. Thus, I employ the Q-learning algorithm, first proposed by Watkins (1989), to address this challenge.

**Learning Equation** The Q-learning algorithm estimates the Q-matrix through an iterative procedure. At the beginning of period $t$, agent $i$ observes $\theta_t$ and thus $s_t$ is determined. Then each agent simultaneously chooses its price $p_{it}$, after which the period payoff $\pi_{it}$ is realized.[14] At the end of period $t$, agent $i$ updates the corresponding cell $(s_t, p_{it})$ in $\mathbf{Q}_{it}$, through the learning equation:

$$Q_{it+1}(s,p) = (1-\alpha) Q_{it}(s,p) + \alpha \left[\pi_{it} + \delta \int_\theta \max_{p' \in A} Q_{it}(s',p') f(\theta)\, d\theta\right] \tag{3}$$

where the new $Q_{it+1}(s,\alpha)$ combines the previous value with the current reward plus the expected discounted value. The weight $\alpha \in [0,1]$ is called the learning rate. Although each agent remembers the last $K$-period prices, the next state $s'$ is uncertain due to randomness

[13] The relationship between the Q-function and the value function is $V(s) \equiv \max_{p \in A} Q(s,p)$.

[14] The action selection rule will be introduced next.

of upcoming demand shocks.[15] A key assumption here is that agents know the distribution of demand shocks, $F(\theta)$. This is reasonable since firms can observe historical demand shocks over enough periods to form accurate beliefs before deploying the pricing algorithms. Thus, given the known distribution of demand shocks, agents can compute the expected discounted value across all possible demand shocks. This stochastic transition represents a major difference from Calvano et al. (2020) and Klein (2021), where the next state $s'$ is deterministic at the end of the current period.

For all other cells $s \neq s_t$ and $p \neq p_{it}$, the Q-value remains unchanged: $Q_{t+1}(s,p) = Q_t(s,p)$. Thus, Q-learning updates only one cell of the Q-matrix at a time.

**Action Selection** The classic $\varepsilon$-greedy rule, which has been effective in approximating the true Q-matrix (Calvano et al., 2020; Klein, 2021), is adopted to determine the price charged by each agent in every period:

$$p_{it} \begin{cases} = \underset{p \in A}{\operatorname{argmax}}\, Q_{it}(s_t, p) & \text{with the prob. } 1-\varepsilon_t \\ \sim \text{Uniform}(A) & \text{with the prob. } \varepsilon_t \end{cases} \tag{4}$$

where $\varepsilon_t = e^{-\beta t}$ is a time-declining exploration rate that effectively balances between exploration and exploitation. In each period, the agent selects either the price that yields the highest Q-value (exploitation mode with probability $1-\varepsilon_t$) or samples a price uniformly at random (exploration mode with probability $\varepsilon_t$).[16]

**Initialization** The initialization of Q-matrix $\mathbf{Q_0}$ possesses degrees of freedom. In the baseline setting, $\mathbf{Q_0}$ is initialized to reflect the fact that at $t = 0$ of each session, each agent operates in full exploration mode ($\varepsilon_0 = 1$) and thus samples uniformly from the discrete price

[15] The state $s_{t+1}$ is $\{\mathbf{p}_{t-k+1}, \ldots, \mathbf{p}_t, \theta_{t+1}\}$, where the last $K$-period prices are remembered but $\theta_{t+1}$ has not yet been realized at period $t$.

[16] Early on, the $\varepsilon$-greedy rule favors exploration to learn about the environment. As learning progresses, it shifts toward exploitation, selecting prices with the highest current Q-values.

space.[17] The alternative initialization is examined in Section 5 for a robustness check.

## 3.3 Parameters

For the baseline economic environment, the market size is $b = 6$ and the marginal cost for each agent is $c_i = 0$. The i.i.d. demand shock $\theta$ takes values in $\{0,4\}$ with equal probability. These two demand states are denoted as low (L) and high (H), respectively, representing the negative and positive demand shocks. Under these conditions, the one-shot Bertrand competition equilibrium price is $p_L^C = p_H^C = 0$, while the monopoly prices are $p_L^M = 3$ and $p_H^M = 5$ for the low and high demand states, respectively.

The theoretical prediction indicates that the cutoff discount factor is $\delta^c = 0.583$. When $\delta > \delta^c$, prices are higher at H than at L. In contrast, when $\delta < \delta^c$, prices reverse across demand states, leading to countercyclical pricing. For details, please see the Appendix D.

The action space $A$ is discretized into 11 equally-spaced points over the interval $[0,5]$, such that $A = \{0, 0.5, \ldots, 5\}$.[18] For computational simplicity, the algorithm employs a one-period memory ($K = 1$). The cardinality of the state space is thus $|S| = 242$. The state at period $t$ is $s_t = (p_{1t-1}, p_{2t-1}, \theta_t)$.

To ensure consistent learning and sufficient exploration, following Calvano et al. (2020), I adopt a learning rate of $\alpha = 0.15$ and an exploration rate of $\beta = 4 \times 10^{-6}$ for the baseline model. A wide range of $\alpha$ and $\beta$ is examined in robustness checks. The discount factor $\delta$ is varied from 0.60 to 0.99 with an increment of 0.01, enabling a comprehensive analysis of the impact of time preferences on pricing behavior. For each parameter configuration $(\alpha, \beta, \delta)$, I conduct $1,000$ simulation sessions, each serving as an independent observation.

A sketch of one simulation session is presented in Algorithm 1 below.

[17] For details, please see the Appendix C.

[18] Under discretization, there exists an additional one-shot symmetric equilibrium—both agents charge the same price of 0.5 across both demand states.

**Algorithm 1** Simulation Procedure for One Session

***First step: Initialization***

$\mathbf{Q}_{i0}$ is generated

***Second step: Loop***

**while** convergence criterion is not satisfied **do**

- $\theta_t$ is realized
- $s_t = (p_{1t-1}, p_{2t-1}, \theta_t)$
- $p_{it}$ is determined through the action selection rule (4)
- $\pi_i(p_{it}, p_{-it}, \theta_t)$ is realized
- $Q_i(s,a)$ is updated through the learning equation (3)

**Single Demand State** To facilitate comparison with the scenario incorporating demand shocks, I also conduct simulations under settings without demand uncertainty: at single demand states L and H, respectively.

**Convergence** While Q-learning convergence in single-agent problems has been proven under certain conditions (Watkins and Dayan, 1992), convergence in strategically interdependent environments remains theoretically unguaranteed. Following Calvano et al. (2020), I employ an empirical convergence criterion: learning is considered complete when each player's optimal strategy remains unchanged for 100,000 consecutive periods. Specifically, convergence is achieved when the optimal price $p_{it}(s) = \underset{p \in A}{\operatorname{argmax}} Q_i(s,a)$ for each player $i$ and state $s$ remains constant over 100,000 consecutive iterations. The simulation terminates upon meeting this criterion or reaching one billion iterations, whichever occurs first. Under the baseline parameters $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$, the average number of iterations required for convergence, across all discount factors and sessions, is 2,331,775 under demand shocks.[19]

[19] In comparison, simulations at single demand states L and H require 1,700,681 and 1,901,235 iterations, respectively. Fewer iterations are expected since the state space at single demand states is half the size of that under demand shocks.

# 4 Results

Building on prior research, I focus on agents' pricing behavior after learning is completed instead of investigating how algorithms converge to their limit strategies. I first explain the methodology for deriving pricing patterns from the limit strategies. Then I evaluate the performance of the pricing patterns in terms of prices and profits and examine their collusive strategies.

## 4.1 Pricing Patterns

### 4.1.1 Limit Strategy

The limit strategy for agent $i$ is the optimal policy derived from the converged Q-matrix, which maps each state to its optimal price, formally defined as

$$p_i^*(s) = \underset{p \in A}{\operatorname{argmax}}\, Q_i(s,p). \tag{5}$$

**Directed Graph** Using the limit strategies $p_1^*(s)$ and $p_2^*(s)$, I construct a directed network to represent the state-price dynamics. Consider a directed graph $G = (V,g)$, where $V = \{1,2,\ldots,M\}$ is the set of nodes. Each node $v = (\theta, p_1, p_2)$ represents a combination of the current demand state and the prices charged by each agent.[20] The adjacency matrix $g$ is an $M \times M$ matrix with each element $g_{ij} \in \{0,1\}$ denoting whether a directed edge exists between nodes $i$ and $j$.[21] A directed edge exists (i.e., $g_{ij} = 1$) in $G$ if and only if the prices in node $v_j$ are optimal responses prescribed by the limit strategies, given the previous prices in node $v_i$ and the current demand state $\theta$ in node $v_j$. Note that it's possible to have $g_{ii} = 1$, which means that if the demand state remains the same from one period to the next, agents do not change their optimal prices, resulting in a self-loop.

[20] Note that $M = 242$.

[21] In a directed network, $g_{ij}$ generally does not equal $g_{ji}$.

### 4.1.2 The Long-run Price Cycle

In this infinitely repeated game, price dynamics are expected to converge to a long-run price cycle $G_c$, which is a subgraph of $G$. In other words, once price dynamics enter $G_c$, all subsequent price movements remain forever in $G_c$.

The long-run price cycle $G_c$ must satisfy two requirements. First, $G_c$ must be a strongly connected component (SCC), meaning that any node in $G_c$ can reach any other node through directed paths.[22] Second, each node in $G_c$ must have all its direct successors within $G_c$. This ensures that random demand shocks cannot cause exits from $G_c$.[23] By construction, $G_c$ is absorbing, as no path exists from any node in $G_c$ to nodes outside it.[24]

Figure 1 illustrates various examples of $G_c$ with different node counts. The label within each node indicates the demand state and the price pair. Directed edges represent transitions between nodes, while an arrow that loops back to the same node represents a self-loop. For example, Figure 1a shows a simple case where $G_c$ contains two nodes. Each node has two outgoing edges: one to the other node and one self-loop. This structure reveals that both agents charge the price of 3 regardless of demand shocks.

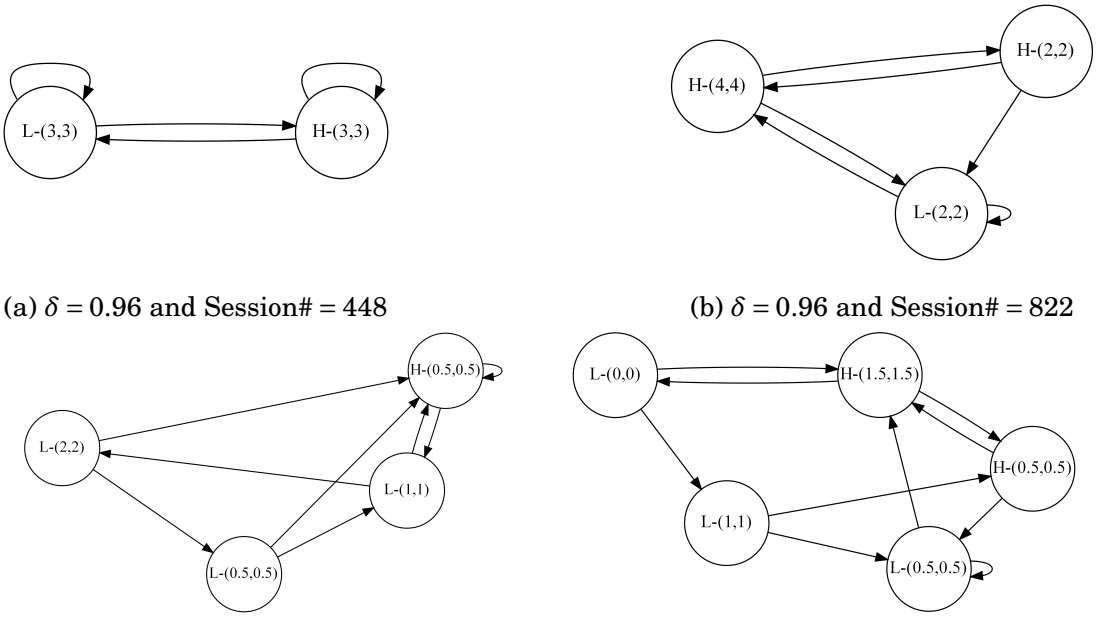
### 4.1.3 Steady-state Price

The price dynamics on $G_c$ follow a finite Markov process. Using its stationary distribution, I calculate the average long-run price at each demand state. The details about derivation are provided in Appendix E. Based on the average long-run prices, I define the pricing patterns as follows.

[22] To identify SCCs in $G$, I employ the algorithm from Tarjan (1972), as modified by Nuutila and Soisalon-Soininen (1994).

[23] The number of outgoing edges for each node equals the number of possible demand states, which is two in the baseline model.

[24] One potential issue is that multiple cycles might exist in $G$. To address this, I drop sessions with multiple cycles. Since 99.94% of sessions contain exactly one cycle, this has virtually no impact on the results. Furthermore, under scenarios restricted to a single demand state (L or H), the frequencies of sessions containing unique cycles remain high at 92.62% and 93.46%, respectively.




(a) $\delta = 0.96$ and Session# = 448

(b) $\delta = 0.96$ and Session# = 822

(c) $\delta = 0.76$ and Session# = 755

(d) $\delta = 0.65$ and Session# = 651

Figure 1: Examples of $G_c$ with Varying Numbers of Nodes

**Definition 1** A long-run price cycle $G_c$ exhibits symmetric and rigid pricing (Sym-Rigid) if it consists of exactly two nodes and the same price is maintained across both agents and demand states.

**Definition 2** A long-run price cycle $G_c$ exhibits procyclical pricing (Pro-Cycle) if the average long-run prices charged by both agents are strictly higher at H than at L.

**Definition 3** A long-run price cycle $G_c$ exhibits countercyclical pricing (Counter-Cycle) if the average long-run prices charged by both agents are strictly higher at L than at H.

The two-node constraint is imposed in the definition of Sym-Rigid because with more than two nodes, agents' prices would necessarily vary across the two demand states, thereby violating price rigidity. For Pro-Cycle and Counter-Cycle, their characterization relies on average long-run prices, independent of the number of nodes. This characterization allows for local price variations while capturing the essential relationship between prices and demand states. Table 1 summarizes the definitions of each pricing pattern.

Table 1: Definitions of Pricing Patterns

| Name | Abbreviation | Pricing |
|---|---|---|
| Symmetric and Rigid Pricing | Sym-Rigid | $p_1^L = p_2^L = p_1^H = p_2^H$ (exactly 2 nodes) |
| Procyclical Pricing | Pro-Cycle | $p_1^H > p_1^L$ and $p_2^H > p_2^L$ |
| Countercyclical Pricing | Counter-Cycle | $p_1^L > p_1^H$ and $p_2^L > p_2^H$ |

Note: $p_i^\theta$ denotes agent $i$'s average long-run price at demand state $\theta$.

#### 4.1.4 Distribution of Pricing Patterns

Figure 2 shows the distribution of pricing patterns: Sym-Rigid, Pro-Cycle, and Counter-Cycle across different values of $\delta$.[25] When $\delta$ is low, Sym-Rigid dominates, with prices close to 0.5. As $\delta$ increases to medium levels (roughly 0.70-0.85), Counter-Cycle emerges as the most prevalent pattern. When $\delta$ is high, Sym-Rigid again becomes the most frequent pattern, though with different price levels. Specifically, when $\delta \geq 0.95$, Sym-Rigid accounts for over 40% of sessions.

The emergence of countercyclical pricing at medium levels of $\delta$ qualitatively aligns with the theoretical prediction. In contrast, at high $\delta$, the predominance of symmetric rigid pricing deviates from theory, which predicts procyclical pricing at high levels of $\delta$. In the following analysis, I first examine Sym-Rigid (at high $\delta$) and then Counter-Cycle (at medium $\delta$) as they represent the most predominant pricing patterns in their respective $\delta$ ranges.

### 4.2 Symmetric Rigid Pricing

#### 4.2.1 Performance

Table 2 summarizes prices and profits under several pricing patterns at $\delta = 0.96$. In Panel A, prices under Sym-Rigid remain constant at 2.66 across both demand states and agents, reaching 89% and 53% of monopoly prices at L and H, respectively. The expected profit across demand states is over 80% of the split monopoly profit (the best collusive outcome).

[25] For detailed price dynamics across $\delta$, see Figure A.2 in the Appendix.

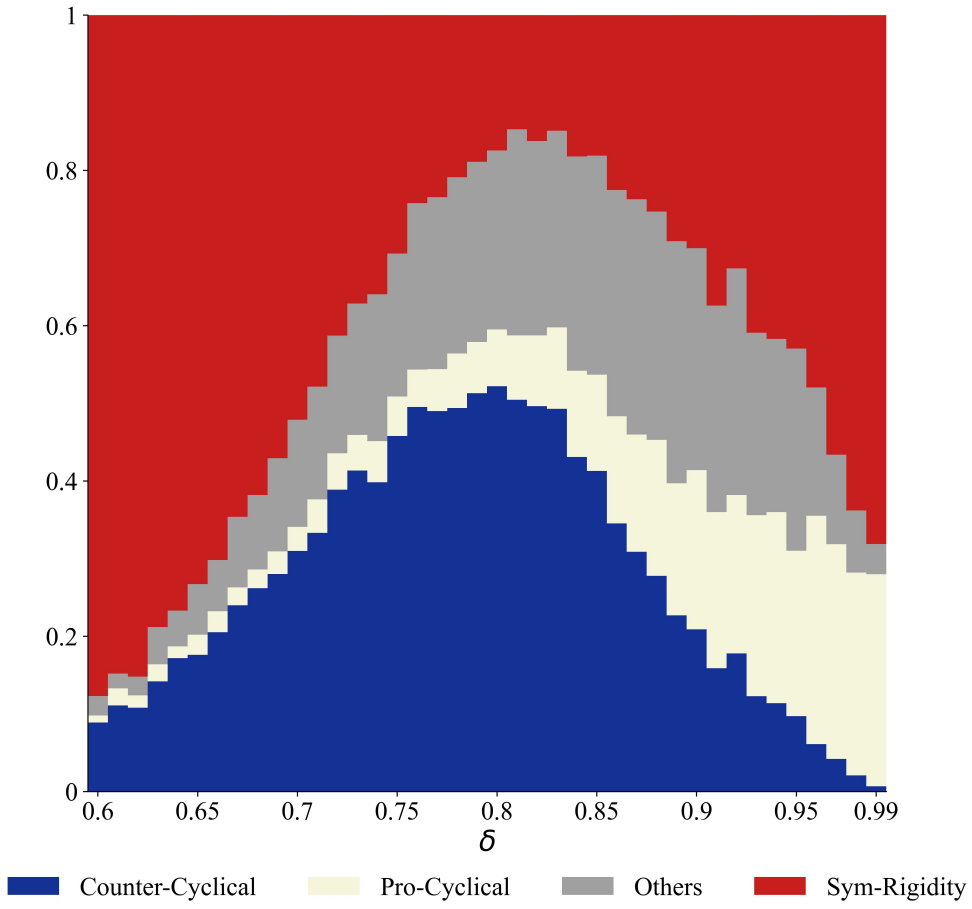



Figure 2: The Distribution of Pricing Patterns across $\delta$

Under Pro-Cycle, the average price at H exceeds that under Sym-Rigid, yet does not lead to higher profit at H. This is because Pro-Cycle exhibits asymmetric pricing in most sessions, where agents charging higher prices lose all market share, earning zero profits. This pricing asymmetry and the lower average price at L jointly reduce the expected profit to 67% of the split monopoly profit, significantly lower than Sym-Rigid (t-test, p-value=0). This profit advantage explains Q-learning's predominant convergence to Sym-Rigid rather than Pro-Cycle (0.49 vs. 0.28).

To further evaluate Sym-Rigid, I examine agents' pricing behavior in the absence of demand shocks. Panel B reports prices and profits for the symmetric pricing pattern, denoted as Sym-1Node, under each single demand state.[26] Under Sym-1Node, agents charge 2.15

[26]The symmetric pricing pattern refers to the long-run price cycle $G_c$ containing exactly one node where

Table 2: Summary Statistics for Prices and Profits when $\delta = 0.96$

| Category | Ratio[a] | $p^L$ | $p^H$ | $\pi^L$ | $\pi^H$ | Expected Profit[b] |
|---|---|---|---|---|---|---|
| *Panel A: Under Demand Shocks* | | | | | | |
| Sym-Rigid | 0.49 | 2.66 | 2.66 | 4.26 | 9.57 | 6.92 |
| | | (0.89[c]) | (0.53) | (0.95) | (0.77) | (0.81) |
| Pro-Cycle[d] | 0.28 | 2.11 | 2.96 | 2.85 | 8.86 | 5.86 |
| | | (0.70) | (0.59) | (0.63) | (0.71) | (0.67) |
| *Panel B: At Single Demand State* | | | | | | |
| Sym-1Node[e] | 0.95, 0.97[f] | 2.15 | 3.09 | 4.04 | 10.48 | 7.26 |
| | | (0.72) | (0.62) | (0.9) | (0.84) | (0.85) |

Notes: a. Frequency of the corresponding pricing pattern among all sessions.
b. Average profit across both demand states.
c. The number in the parentheses denotes the proportion relative to the monopoly price or the split monopoly profit.
d. Pro-Cycle may involve asymmetric pricing. However, as average prices and profits are nearly identical for both agents, Panel A shows only agent 1's results.
e. Sym-1Node denotes the symmetric pricing pattern at the single demand states, where $G_c$ contains exactly one node and both agents charge the same price.
f. These two numbers refer to the frequencies of Sym-1Node at the single demand states L and H, respectively.

at L and 3.09 at H. In comparison, Sym-Rigid's constant price of 2.66 lies between them, smoothing prices across demand states. This price smoothing yields an expected profit comparable to the average profit across both single demand states (0.81 vs. 0.85, relative to the split monopoly profit).

Figure A.3 in the Appendix compares prices and profits between Sym-Rigid and Sym-1Node. As $\delta$ increases, the expected profit under Sym-Rigid progressively approaches the average profit under Sym-1Node (across the two single demand states), converging to the upper bound obtained in the economic environment without exogenous demand shocks. Notably, at $\delta = 0.99$, Sym-Rigid's average price rises to 3.31, even exceeding the monopoly price at L, and its expected profit is 7.55, which constitutes 96.3% of the average profit from the two single demand states.

**Result 1:** When $\delta$ is high, Sym-Rigid emerges as the predominant pricing pattern. It

both agents charge the same price.

occurs more frequently than Pro-Cycle and generates supra-competitive profits. Compared to scenarios with single demand states, Sym-Rigid smooths prices across demand states, and its expected profit approaches the upper bound as $\delta$ increases.

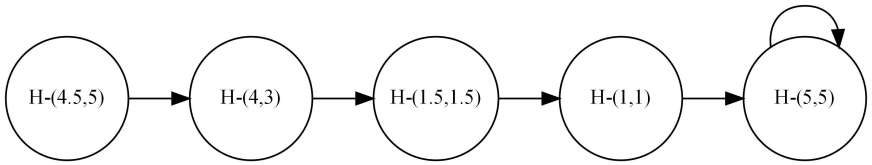
### 4.2.2 Analysis of Collusion

Although Sym-Rigid can maintain supra-competitive profits under high $\delta$, this alone is not sufficient to conclude collusion. As defined by Harrington (2018), collusion occurs when firms use strategies that incorporate a reward–punishment scheme, which rewards a firm for abiding by the supra-competitive outcome and punishes any deviation from it. While each agent's strategy can be derived from the Q-matrix, as noted by Calvano et al. (2020), fully describing these strategies is challenging due to the numerous contingencies and the variation in strategies across different sessions. Therefore, to address this challenge, I follow the method proposed by Calvano et al. (2020) to conduct the deviation test.

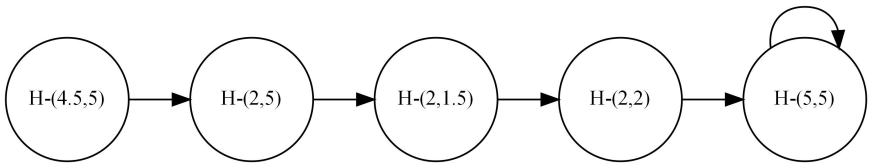
However, the presence of (observed) demand shocks complicates the deviation test. Without demand shocks, each price undercut leads to a unique and deterministic deviation path. However, with demand shocks, the uncertainty in demand states leads to multiple possible deviation paths following a price undercut. To illustrate, Figure 3 shows the two single-path deviations at the single demand state H, while Figure 4 presents a multi-path deviation under demand shocks when undercutting occurs at H. In Figure 4, the stable price in $G_c$ is 4 across both demand states and agents, represented by the two rightmost nodes with self-loops and mutual arrows. The leftmost node shows the initial deviation where agent 1 undercuts by 0.5. The intermediate nodes and arrows form multiple possible paths from the initial deviation back to $G_c$.[27]

To address this complication, I conduct the deviation test through one thousand simulations and then calculate averages, keeping constant the level of price undercut, the identity of the deviating agent, and the initial deviating demand state. In each deviation simulation,

[27] For instance, one deviation path is $\mathrm{H}-(3.5,4) \to \mathrm{H}-(1.5,1.5) \to \mathrm{H}-(1,1) \to \mathrm{L}-(4,4)$, with a length of 4.




(a) Profitable Deviation Path (Session# = 911)




(b) Unprofitable Deviation Path (Session# = 253)

Figure 3: Two Single-path Deviations at Single Demand State H

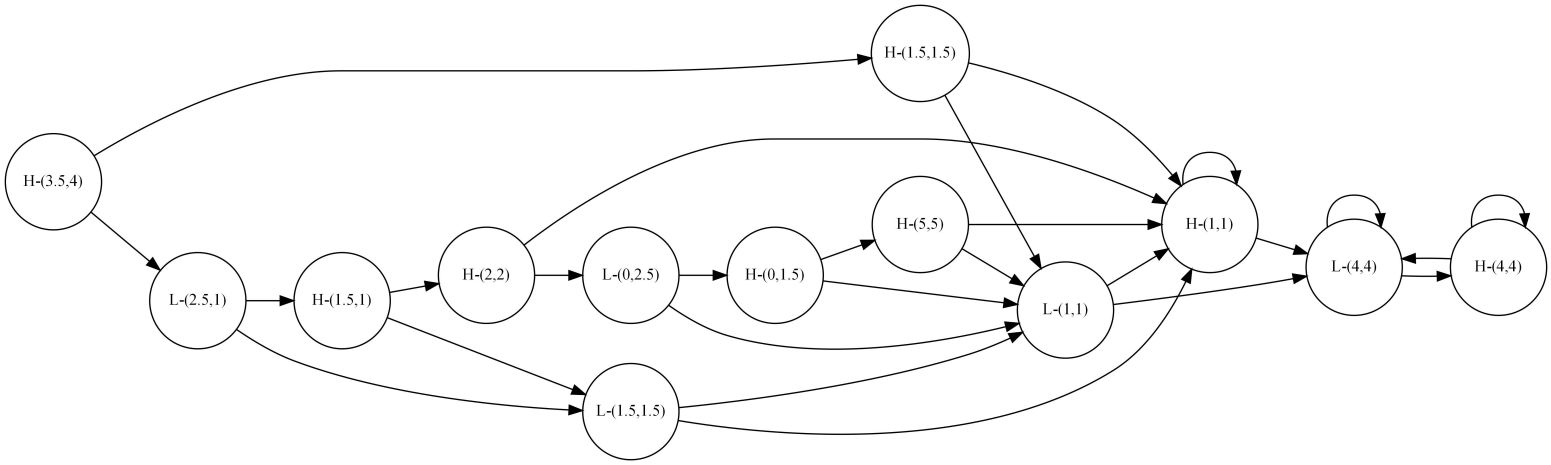



Figure 4: A Multi-path Deviation under Demand Shocks (Deviation Occurs at H and Session# = 513)

I force one agent to depart from $G_c$ by undercutting the stable price in the initial period, while the other agent maintains the stable price (unilateral deviation). Following the initial deviation, both agents set prices according to their limit strategies until the price dynamics return to $G_c$. Starting from the initial deviation and continuing until the price dynamics return to $G_c$, I record both the actual profits on the deviation path and the counterfactual profits that would have been earned if both agents stayed in $G_c$ (non-deviation path). For each agent, the accumulated discounted profit on each path is averaged across one thousand simulations, treated as an independent result. I compare the profits from deviation

and non-deviation paths to evaluate whether the deviation is profitable. Algorithm 2 in the Appendix provides a detailed illustration of the deviation simulation procedure.

Table 3 summarizes results from deviation tests when agent 1 undercuts by the smallest price unit (i.e., 0.5), which generates the highest initial deviating profit in most cases.[28] Panel A shows that under Sym-Rigid, the frequencies of profitable deviations (denoted as $q^D$) are 0.17 and 0.31 when deviation happens at L and H, respectively, implying that deviations are unprofitable approximately 80% and 70% of the time. The higher frequency when deviation happens at H stems from the larger initial profit gain from undercutting. The deviating agent's profit ratios (deviation path relative to non-deviation path, denoted as $\%\Pi_1$) are 0.81 and 0.91 for L and H, respectively. Thus, both measures—the low frequencies of profitable deviations and the profit ratios below one—indicate that such price undercutting is unprofitable.

Since the above deviation analysis only focuses on the smallest price unit deviation, I further examine all possible undercutting scenarios. Table A.1 in the Appendix details the frequencies of profitable deviations for the deviating agent at each possible price undercut. These frequencies are uniformly low, and they rapidly decrease to zero as the magnitude of price undercutting increases. Additionally, Table A.2 in the Appendix presents the profit ratios (deviation path relative to non-deviation path) for each possible price undercut, with all ratios below one. Taken together, these results demonstrate that deviations are unprofitable regardless of the extent of price undercutting.

Last, an interesting pattern emerges when comparing scenarios with and without demand shocks. Panel B in Table 3 presents the results of deviation tests at each single demand state.[29] When deviations occur at L, profitable deviations are less frequent with

[28]The results of deviation tests remain nearly identical regardless of which agent initiates the deviation. Therefore, for illustration purposes, I present the results from forcing agent 1 to undercut.

[29]Deviation tests also show that deviations are unprofitable at each single demand state. For results of all possible undercuts, see Panel B in Table A.1 and Table A.2 in the Appendix. The frequencies of profitable deviations without demand shocks are higher than in Calvano et al. (2020), primarily because this paper assumes homogeneous products, while Calvano et al. (2020) incorporates product differentiation. With homogeneous products, competition is more intense as agents who undercuts can capture the entire market, leading to higher potential profits from deviating.

Table 3: Summary Statistics for Deviation Tests when $\delta = 0.96$

| Agent 1 undercuts by 0.5 | Deviation occurring at L | | | | Deviation occurring at H | | | |
|---|---|---|---|---|---|---|---|---|
| Category | length[a] | $q^D$ (freq.)[b] | $\%\Pi_1$ (deviator)[c] | $\%\Pi_2$ (non-deviator)[d] | length | $q^D$ (freq.) | $\%\Pi_1$ (deviator) | $\%\Pi_2$ (non-deviator) |
| *Panel A: Under Demand Shocks* | | | | | | | | |
| Sym-Rigid | 7.54 | 0.17 | 0.81 | 0.55 | 7.56 | 0.31 | 0.91 | 0.47 |
| *Panel B: At Single Demand State* | | | | | | | | |
| Sym-1Node | 4.53 | 0.24 | 0.89 | 0.49 | 4.91 | 0.21 | 0.87 | 0.47 |

Notes: Panel A displays results from deviation tests under demand shocks; Panel B presents those results at each single demand state, L and H.
a. Average periods for price dynamics returning to $G_c$.
b. Frequency of profitable deviations.
c. The deviating agent's profit ratio: deviation path relative to non-deviation path.
d. The non-deviating agent's profit ratio: deviation path relative to non-deviation path.

demand shocks than without (0.17 vs. 0.24, t-test, p-value = 0). Conversely, when deviations occur at H, profitable deviations are more frequent in the presence of demand shocks (0.31 vs. 0.21, t-test, p-value = 0). The intuition is straightforward: When deviations occur at L, future losses occur at both demand states, making the expected losses larger than at single demand state L. When deviations occur at H, however, future losses are spread across both demand states, which mitigates the total expected losses.

**Result 2:** The deviation tests show that across all possible undercuts, expected gains from deviation are outweighed by expected losses, and profitable deviations remain unlikely (for both agents). Thus, the collusive outcomes under Sym-Rigid are sustained. This unprofitability indicates the existence of punishment implemented by the non-deviating agent, which effectively suppresses the deviating agent's profits along the deviation path.

**Features of Deviation Path** Following the deviation test, I summarize three distinct patterns of agent pricing behavior on the deviation path.

*Probabilistic profitability of deviations.* Under demand shocks, multi-path deviations make the profitability of a deviation probabilistic, determined by the occurrence rates of profitable paths, rather than deterministic (0 or 1) at single demand states. For example, in Figure 4, two profitable deviation paths have a combined occurrence rate of 18.75%.[30] In

[30]These two profitable deviation paths are $H-(3.5,4) \to H-(1.5,1.5) \to L-(1,1) \to L-(4,4)$ and $H-(3.5,4) \to$

contrast, Figure 3 shows the deterministic case of two deviation paths at the single demand state H, where one path is profitable and the other is not.

*Frequent low-price transitions.* The second key feature of the deviation path is the frequent occurrence of specific low price pairs before returning to $G_c$. To quantify their prevalence, I employ the in-degree centrality metric, which measures how often these price pairs are visited.[31] Figure 5 displays the average in-degree centrality for each price pair following agent 1's undercut by the minimum price unit after observing a positive demand shock. The deepest color blocks in the heat maps are concentrated in price ranges from 0.5 to 1.5, indicating these price pairs serve as common transition nodes. These transition nodes play an crucial role in punishment, substantially contributing to the unprofitability of deviations. Notably, one of the two symmetric one-shot equilibria—$(0,0)$ at both demand states—is rarely used as punishment, possibly because agents recognize that charging the price of 0, while harming their rival, generates no profit for themselves.[32] The pattern of frequent low-price transitions remains consistent across different undercut levels, deviating agents, pre-deviation prices, and initial deviating demand states. Figure A.4 in the Appendix presents the average in-degree centrality for deviation initiating at L, which is almost identical to Figure 5.

*Discontinuous switching pattern.* The price dynamics on the deviation path exhibit a discontinuous punishment-cooperation switching pattern: upon detecting a deviation, both agents enter a punishment phase, charging low prices regardless of demand states. This phase persists for several periods before abruptly returning to the cooperation phase ($G_c$). Figure 6 illustrates a representative example of how prices return to $G_c$ following a uni-

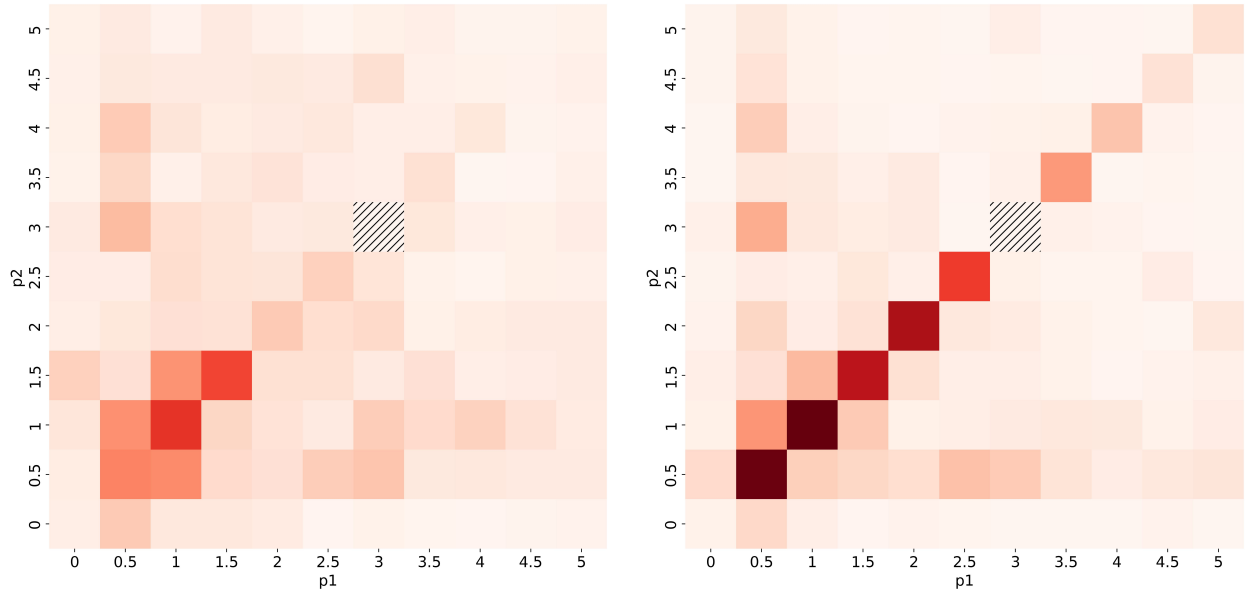
$\mathrm{L}-(2.5,1) \to \mathrm{L}-(1.5,1.5) \to \mathrm{L}-(1,1) \to \mathrm{L}-(4,4)$, with occurrence rates of 12.5% and 6.25%, respectively.

[31] The in-degree centrality for a node $v$ in a directed graph is calculated as:

$$C_{D_{\text{in}}}(v) = \frac{D_{\text{in}}(v)}{V-1}$$

where $D_{\text{in}}(v)$ is the number of edges entering node $v$ and $V$ is the total number of nodes in the graph. The initial deviating nodes are excluded when calculating the in-degree centrality.

[32] Moreover, those low price pairs at H generally show deeper colors than at L, suggesting higher transition intensity at H.




(a) Price Pairs at L

(b) Price Pairs at H

Figure 5: In-Degree Centrality of Price Pairs on the Deviation Path (Deviation Initiating at H)

Notes: The square depicted in lines marks the pre-deviation price of 3. A deviation initiates when agent 1 undercuts by the minimum price unit after observing a positive demand shock. The color gradient indicates the level of in-degree centrality.

lateral deviation. In the cooperation phase ($G_c$), both agents charge 3 across both demand states. At $t = 1$, given a positive demand shock, the deviating agent undercuts by 0.5. At $t = 2$, the non-deviating agent responds with punishment by dropping its price to a low level. The deviating agent, anticipating this punishment, also charges low prices. Both agents maintain low prices regardless of demand states during the punishment phase, before prices quickly return to $G_c$.[33] Compared with the punishment pattern identified in Calvano et al. (2020), the similarity is that this discontinuous switching pattern also resembles the "stick-and-carrot" strategies of Abreu (1986).[34] However, a major distinction is that in Calvano et al. (2020), prices gradually return to pre-deviation levels without demand shocks. This distinction is economically intuitive: in stochastic environments with

[33]Further analyses confirm that this pattern persists under Sym-Rigid. Figure A.5 in the Appendix displays the deviating agent's price dynamics under various lengths of deviation paths.

[34]The departure is that the punishment is not as harsh as stated in Abreu (1986) (Bertrand price).

fluctuating demands, algorithms struggle to learn and maintain stable gradual paths back to cooperation. Instead, through learning, they develop clear "restart points" that, once reached, trigger swift returns to cooperative high prices.

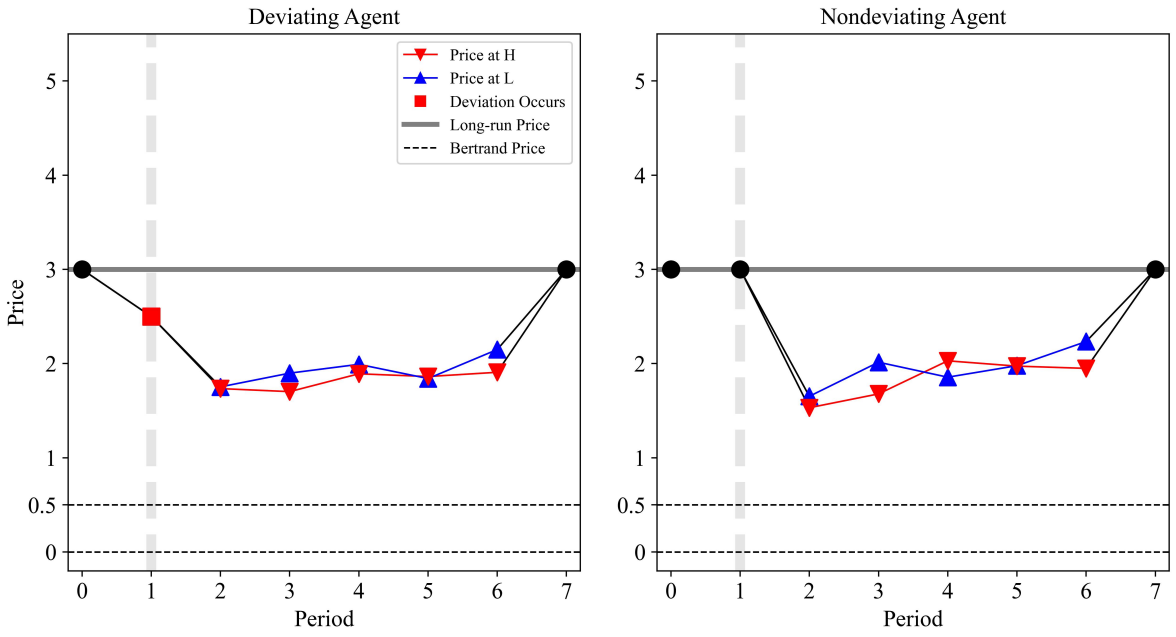



Figure 6: Price Dynamics after Deviation

Notes: The length of the deviation path is fixed as 7, where at period $t = 0$ and 7, the price equals the stable price in $G_c$. The prices for each demand state at each period are averaged across all Sym-Rigid sessions. The deviation initiates at H when $t = 1$.

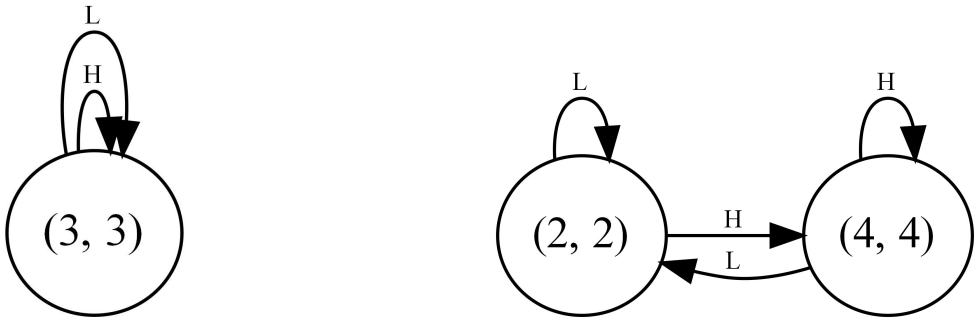
**Result 3:** Multi-path deviations under Sym-Rigid exhibit three key features: probabilistic profitability of deviations, frequent low-price transitions, and discontinuous punishment-cooperation switching pattern. The last feature is particularly notable: once a deviation is detected, both agents switch to charging low prices regardless of demand states. This punishment phase lasts for several periods until restart points are reached, after which agents quickly return to cooperation.

### 4.2.3 Discussion

So far, we have examined the performance in terms of prices and profits and features of the deviation path under Sym-Rigid. An unanswered question is why the predominant pricing

pattern is Sym-Rigid rather than Pro-Cycle. This contradicts the theoretical prediction by Rotemberg and Saloner (1986) that agents would charge higher prices during higher demand states at sufficiently high $\delta$. Since Sym-Rigid accounts for about half of all sessions ($\delta = 0.96$), a more specific question arises: why do algorithms predominantly learn rigid pricing rather than procyclical pricing when constrained to two-node graph structures?

One plausible explanation stems from agents' limited memory: recall that they only remember the previous period's price pair, not the accompanying demand state. This algorithmic design reduces the state space dimension while causing agents to treat identical price pairs from different demand states as equivalent, making Sym-Rigid more likely to emerge. To illustrate, consider two examples: a symmetric rigid pricing pattern $L-(3,3)$ and $H-(3,3)$, and a procyclical pricing pattern $L-(2,2)$ and $H-(4,4)$. Figure 7 represents these two pricing patterns by finite automata. The rigid pricing pattern is characterized by a single state (price pair) with two self-referencing transitions. In contrast, the procyclical pricing pattern requires two states and four transitions: two inter-state and two intra-state, as it features different price pairs across demand states. The lower complexity of rigid pricing, both in terms of states and transitions, helps explain why Q-learning algorithms more readily discover and implement this pricing pattern.




(a) Automaton for Sym-Rig (b) Automaton for Pro-Cycle

Figure 7: Pricing Patterns Represented by Finite Automata

Following this argument, if agents can remember both the previous period's price pair and demand state, then identical price pairs under different demand states become distin-

guishable (e.g., $L-(3,3)$ and $H-(3,3)$ are treated differently). Consequently, procyclical pricing should become the predominant pricing pattern for sufficiently high $\delta$ as generally procyclical pricing can create more profits. The simulation results confirm this prediction.[35] Constrained to two-node graph structures, procyclical pricing appears in 24.2% of all sessions, while rigid pricing accounts for only 2.3%. More broadly, procyclical pricing represents 57.3% of all sessions, demonstrating that the information contained in state variables significantly shapes agents' learning outcomes.

## 4.3 Countercyclical Pricing

Next, I analyze the predominant pricing pattern in the medium range of $\delta$, which is countercyclical pricing (Counter-Cycle).

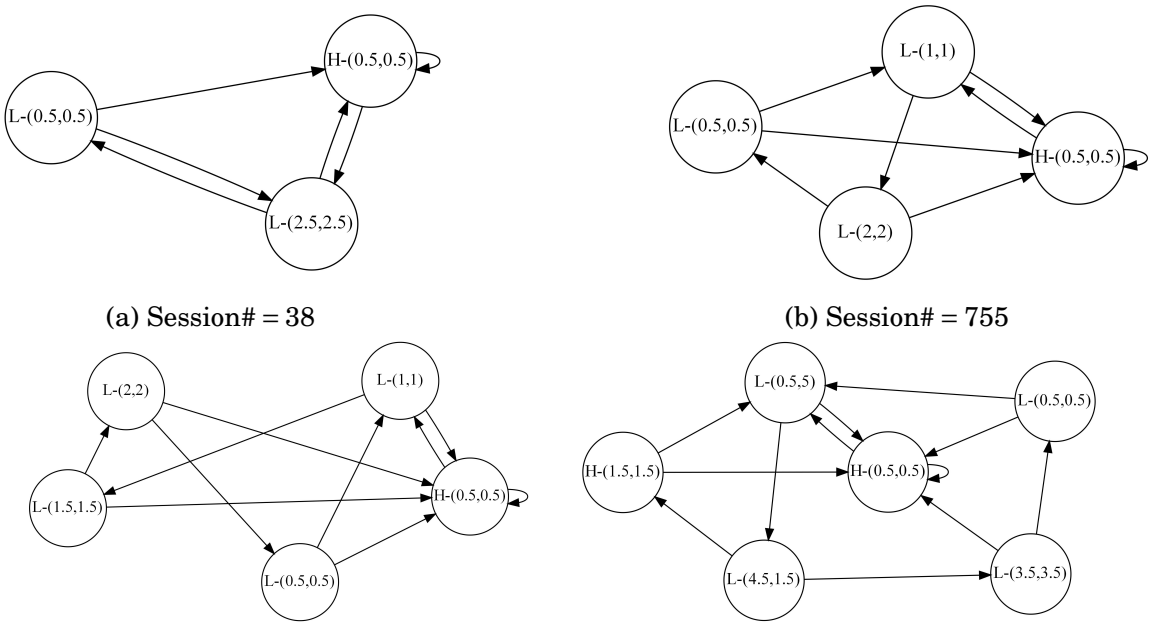
### 4.3.1 Graph Structure

Figure 8 illustrates several examples of long-run price cycles $G_c$ under Counter-Cycle when $\delta = 0.76$. In contrast to Sym-Rigid, which has exactly one node per demand state, $G_c$ under Counter-Cycle typically exhibits multiple nodes at each demand state. As shown in Table 4, when $\delta = 0.76$, the average number of nodes under Counter-Cycle is 5.49 at L and 3.58 at H. Thus, Counter-Cycle is characterized by a larger and more complex graph structure. Although a comprehensive characterization of the graph structure of $G_c$ remains challenging, one thing clearly stands out: the node $H-(0.5,0.5)$, which is one of the symmetric one-shot equilibria at H, plays a pivotal role in contributing to the average lower price observed at H. Three key features of $H-(0.5,0.5)$ are summarized below.[36]

*High frequency.* As shown in Table 4, the frequency of $H-(0.5,0.5)$ is 99.8%, appearing in almost every $G_c$. In stark contrast, the frequency of $L-(0.5,0.5)$ diminishes significantly to 66.9%.

[35] When agents remember the previous period's demand state, the state space doubles compared to the baseline setting. Considering this, the chosen parameters are $\delta = 0.15$, $\alpha = 0.15$, and $\beta = 1 \times 10^{-6}$. The smaller value of $\beta$ prolongs exploration before convergence.

[36] These three key features of $H-(0.5,0.5)$ are robust throughout the medium range of $\delta$.




(a) Session# = 38 (b) Session# = 755

(c) Session# = 187 (d) Session# = 960

Figure 8: Representative Examples of $G_c$ under Counter-Cycle with $\delta = 0.76$

Table 4: Summary Statistics for $G_c$ under Counter-Cycle with $\delta = 0.76$

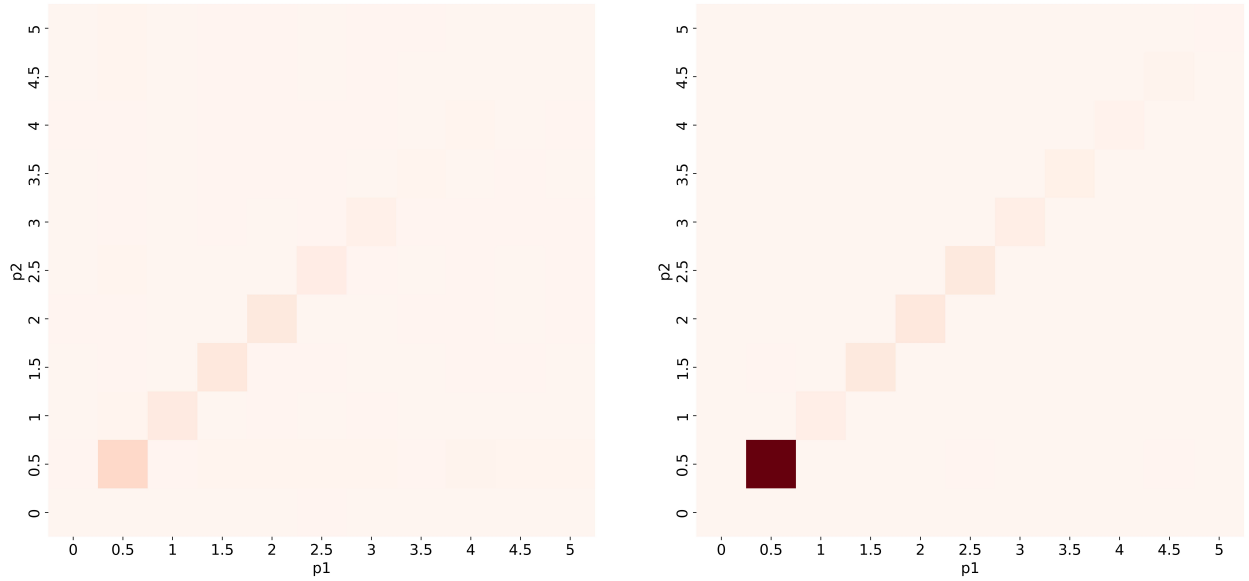
| Demand State | Obs. | Size[a] | $p_1$[b] | $p_2$ | Freq. of $(0.5, 0.5)$[c] |
|---|---|---|---|---|---|
| L | 495 | 5.49 | 2.12 | 2.09 | 0.669 |
| H | 495 | 3.58 | 1.06 | 1.05 | 0.998 |

Notes: a. The average number of nodes of $G_c$ at each demand state.
b. The average price of agent 1 at each demand state.
c. The frequency of the price pair $(0.5, 0.5)$ at each demand state.

*High centrality.* Figure 9 visualizes the in-degree centrality for price pairs at each demand state when $\delta = 0.76$. $H-(0.5,0.5)$ stands out with an exceptionally high in-degree centrality, represented by the darkest color block, while all other price pairs show much lighter colors. This visual contrast underscores its uniquely high degree of connectivity within $G_c$. Simulation tests corroborate such connectivity, showing that starting from any node in $G_c$, it takes on average less than two periods to reach $H-(0.5,0.5)$.

*High self-loop tendency.* Among all price pairs at both demand states, $H-(0.5,0.5)$ has the highest self-loop frequency at 85.6%, significantly exceeding any other price pair. This characteristic suggests that price dynamics exhibit a strong tendency to persist at

$H-(0.5,0.5)$, conditional on the positive demand shock occurring in the next period.




(a) Price Pairs at L

(b) Price Pairs at H

Figure 9: In-Degree Centrality of Price Pairs within $G_c$ under Counter-Cycle

Notes: The color gradient in the figure indicates the level of in-degree centrality, with darker colors representing higher centrality.

**Result 4:** Under Counter-Cycle, $H-(0.5,0.5)$ appears in nearly every $G_c$ and acts as a strong attractor in the price dynamics, causing other nodes to reach it in fewer than two periods and then remain there (conditional on the positive demand shock in the next period). These features persist throughout the medium range of $\delta$, substantially contributing to the lower average prices at H and creating very limited opportunities for agents to make profitable deviations at H.

### 4.3.2 Performance

Then I examine the performance of Counter-Cycle in terms of prices and profits. Table 5 presents summary statistics for several pricing patterns at $\delta = 0.76$. Under Counter-Cycle, the average price at L is 2.11, which is 70% of the monopoly price, yielding 62% of the split monopoly profit. This indicates that supra-competitive profit is maintained during the

negative demand shocks. In contrast, at H, the average price is only 1.05, representing 21% of the monopoly price and yielding 32% of the split monopoly profit. Overall, the profit under Counter-Cycle, averaged across both demand states, attains 40% of the best collusive outcome. While this might appear low, Counter-Cycle still maintains 75.4% of the average profit achieved across two single demand states (shown in Panel B).

Table 5: Summary for Prices and Profits when $\delta = 0.76$

| Category | Ratio | $p^L$ | $p^H$ | $\pi^L$ | $\pi^L$ | Expected Profit |
|---|---|---|---|---|---|---|
| *Panel A: Under Demand Shocks* | | | | | | |
| Sym-Rigid | 0.24 | 0.67 | 0.67 | 1.71 | 3.04 | 2.37 |
| | | (0.22) | (0.13) | (0.38) | (0.24) | (0.28) |
| Counter-Cycle | 0.49 | 2.11 | 1.05 | 2.77 | 3.98 | 3.38 |
| | | (0.70) | (0.21) | (0.62) | (0.32) | (0.40) |
| *Panel B: At Single Demand State* | | | | | | |
| Sym-1Node | 0.95, 0.71 | 0.76 | 1.77 | 1.89 | 7.06 | 4.48 |
| | | (0.25) | (0.35) | (0.42) | (0.57) | (0.53) |

Notes: The settings are identical to those in Table 2. As prices and profits are nearly identical for agents 1 and 2, this table presents results solely for agent 1.

In contrast, under Sym-Rigid, the fixed price of 0.67 is lower than the prices observed under Counter-Cycle for both demand states. Consequently, the expected profit under Sym-Rigid is significantly lower than that achieved under Counter-Cycle (2.37 vs. 3.38).[37] Table 5 further shows that Counter-Cycle emerges in nearly half of all sessions, appearing twice as frequently as Sym-Rigid. Counter-Cycle's superior performance rationalizes its emergence as the predominant learning outcome. Figure A.6 in the Appendix compares the profits of Counter-Cycle and Sym-Rigid across $\delta$. The comparison reveals a critical threshold at $\delta \approx 0.8$. Below this threshold, Counter-Cycle exhibits superior performance, while Sym-Rigid takes the lead above it. These results indicate that Counter-Cycle generally outperforms Sym-Rigid when $\delta$ is not sufficiently high.

**Result 5:** In the medium range of $\delta$, Counter-Cycle emerges as the predominant pricing pattern. Under this pricing pattern, the Q-learning algorithms successfully coordinate to

[37] The p-value of the t-test is 0.

maintain supra-competitive profits at L but exhibit poor performance at H, achieving relatively low expected profits. Nevertheless, Counter-Cycle still significantly outperforms Sym-Rigid, demonstrating that its emergence as the predominant learning outcome is not a random result.

### 4.3.3 Analysis of Collusion

The deviation tests are necessary for evaluating collusion under Counter-Cycle, as they were previously done for Sym-Rigid. However, within the long-run price cycle $G_c$ under Counter-Cycle, the key features of the node $H-(0.5,0.5)$—particularly its frequent occurrence and traversal—may impede the effective implementation of punishment.

From the theoretical perspective, Barlo et al. (2009) argue that the presence of Nash reversion on the equilibrium path may disrupt coordination under bounded memory. Specifically, agents cannot distinguish between states on the equilibrium path and states off the equilibrium path where punishment (Nash reversion) is occurring. Following this argument, given the features of $H-(0.5,0.5)$ within $G_c$, when an agent deviates and price dynamics shift to the deviation path, agents are likely to choose the one-shot symmetric equilibrium $H-(0.5,0.5)$ as punishment during the positive demand shocks.[38] Due to one-period bounded memory, the price dynamics subsequently return to $G_c$. As a result, the duration of the deviation path may not be sufficiently long, potentially making the punishment ineffective and thus the deviation profitable. The deviation tests under Counter-Cycle support this prediction, revealing that deviations initiating from either demand state are profitable with a frequency of approximately 55%.

Then a puzzle emerges: despite the relatively high frequency of profitable deviations, why does the countercyclical pricing pattern still predominantly hold? Two factors help explain this phenomenon. First, given the features of $H-(0.5,0.5)$, the deviation paths are shortened: when the algorithms engage in random play (exploration mode) and depart from

[38]The other symmetric one-shot equilibrium, $H-(0,0)$, is not likely to be chosen as it yields strictly zero profit for both agents, confirmed before in the deviation tests under Sym-Rigid.

$G_c$, price dynamics would swiftly return to $G_c$. Consequently, with more time spent in $G_c$, Q-values accumulate more frequently for the nodes within $G_c$ compared to those outside it. Second, the presence of multiple nodes within $G_c$ helps dilute the frequency of deviations occurring during exploration mode at any single node. Together, these two factors ensure that the structure of $G_c$ remains unchanged.[39]

**Result 6:** Counter-Cycle does not pass the deviation test. This failure is primarily due to the frequent occurrence and traversal of the node $H-(0.5,0.5)$ within $G_c$, which impedes agents from effectively utilizing it as a form of punishment on the deviation path. Nevertheless, this pricing pattern persists, attributed to two factors: more frequent Q-value accumulation within $G_c$ due to shortened deviation paths, and the presence of multiple nodes that disperse deviations during exploration.

#### 4.3.4 Discussion

Figure 2 shows that Sym-Rigid prevails at both high and low values of $\delta$, while Counter-Cycle emerges as the predominant pricing pattern and significantly outperforms Sym-Rigid in the medium range of $\delta$. These findings suggest that the emergence of Counter-Cycle is not a random outcome, but rather represents an optimal response to these specific levels of time preference, as captured by $\delta$. A natural question then arises: how do the Q-learning algorithms learn to form countercyclical pricing patterns? Specifically, how do they depart from rigid pricing and learn to charge lower prices during the positive demand shocks?

The explanation is reminiscent of Rotemberg and Saloner (1986). Starting from Sym-Rigid, as $\delta$ decreases to the medium range, agents through exploration mode gradually learn that deviating during the positive demand shocks becomes more profitable. Consequently, undercutting at H becomes increasingly frequent. In response, the non-deviating agent lowers its price, resulting in a price war. The resulting downward pressure on pricing leads agents to set increasingly lower prices. Eventually, this process stabilizes with

[39]For reference, Cho and Williams (2024) analytically show that after shutting down all collusion channels, collusive outcomes can still be sustained when endogenizing the algorithmic selection.

consistently low prices at H, where playing $H-(0.5,0.5)$ frequently within $G_c$ serves as a form of self-punishment, eliminating the potential gains from deviating at H. Meanwhile, during the negative demand shocks, since $\delta$ is not sufficiently low, the deviating profits do not outweigh the gains from cooperation (stay in $G_c$). Consequently, the higher prices at L can be maintained. This asymmetric pricing behavior across demand states leads to the emergence of countercyclical pricing, which represents an adaptive response to the specific time preference.

While the Q-learning algorithms do not fully achieve the optimal performance predicted by Rotemberg and Saloner (1986), they still learn to implement countercyclical pricing, giving support to the theoretical foundation for this pricing pattern. As $\delta$ further decreases to the low range, the relatively high prices at L become unsustainable as the incentive to deviate strengthens, and Sym-Rigid—characterized by charging the fixed low prices across both demand states and agents—emerges as the predominant pricing pattern once again.

# 5 Robustness Check

In this section, I examine whether the findings from the baseline setting—specifically the identified pricing patterns and tacit collusive outcomes—are robust to various parameter changes. Furthermore, I explore how agents respond to variations in their ability to observe demand shocks.

## 5.1 Alternative Initialization

To examine the robustness of the identified pricing patterns to alternative Q-matrix initialization, I set the initial Q-values to zero for both agents. The emerging pricing patterns remain consistent with previous findings. Specifically, under the baseline parameters $\alpha = 0.15$ and $\beta = 4 \times 10^{-6}$, Sym-Rigid appears with a frequency of 0.59 at $\delta = 0.96$, while Counter-Cycle emerges with a frequency of 0.45 at $\delta = 0.76$.

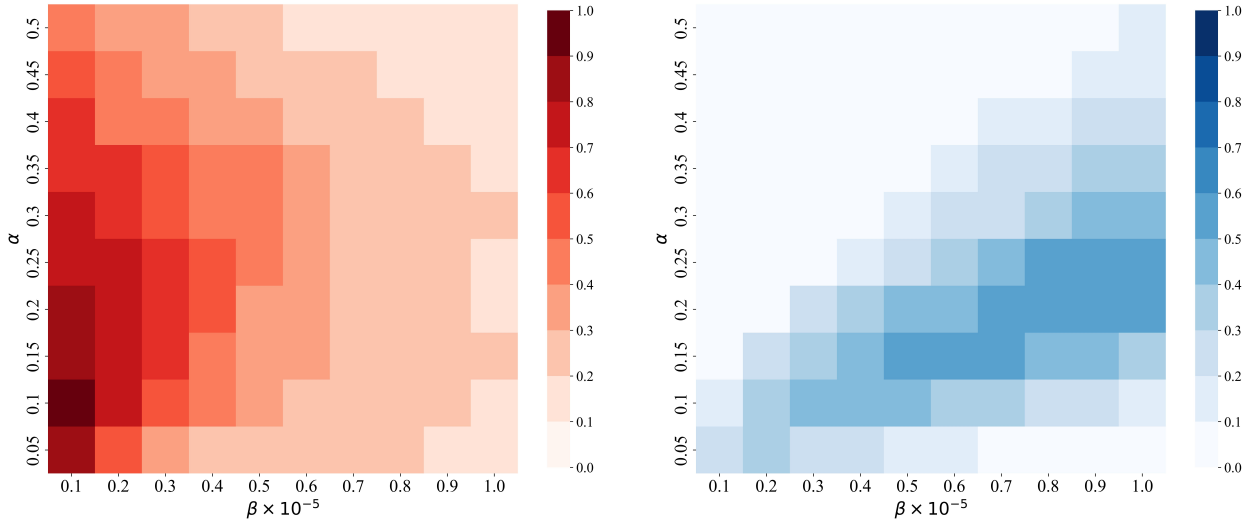
## 5.2 Parameter Variations

The learning rate $\alpha$ and the experimentation parameter $\beta$ are treated as exogenous variables. I systematically vary these parameters to examine whether the main findings are robust across different parameter values or merely specific to particular parameter choices. For both parameters, I establish grids of 10 equally spaced points: $\alpha \in [0.05, 0.5]$ and $\beta \in [10^{-6}, 10^{-5}]$. The range of $\alpha$ spans different balances between incorporating new information and maintaining past learning, while the range of $\beta$ ensures various levels of exploration during learning.

**Pricing Patterns** Figure 10 presents the frequencies of pricing patterns across the parameter space of $\alpha$ and $\beta$. In Panel A, Sym-Rigid ($\delta = 0.96$) emerges as the predominant pricing pattern, with its occurrence frequency positively correlated with exploration rates (i.e., lower $\beta$). Notably, for low $\alpha$ and $\beta$, the frequency exceeds 70%, demonstrating this pricing pattern's persistent dominance within this parameter subspace. Panel B displays the frequency of Counter-Cycle with $\delta = 0.76$ across parameter combinations. The prevalence of Counter-Cycle is lower than Sym-Rigid shown in Panel A, with frequencies generally below 60%. This is because when $\delta$ is in the medium range, Counter-Cycle and Sym-Rigid compete for dominance, with many sessions ultimately converging to Sym-Rigid.[40] Nevertheless, Counter-Cycle still emerges consistently across a substantial range of parameters. The results in both panels show that the two predominant pricing patterns, Sym-Rigid at high $\delta$ and Counter-Cycle at medium $\delta$, are robust and not artifacts of particular parameter choices.

**Collusive Profits** Figure 11 presents the expected profits under Sym-Rigid ($\delta = 0.96$) as a function of $\alpha$ and $\beta$. Across the parameter space, profits range from 65% to 90% of the best collusive outcome and show little sensitivity to changes in the learning and experimentation parameters. The deviation tests also show that deviations are not profitable across

[40] The predominant pricing pattern in the upper triangle with the lightest color of Panel B is Sym-Rigid.




(a) Sym-Rigid with $\delta = 0.96$ (b) Counter-Cycle with $\delta = 0.76$

Figure 10: Frequencies of Two Pricing Patterns across Parameter Space

this parameter range. These results demonstrate the robustness of both supra-competitive profits and tacit collusion under Sym-Rigid when $\delta$ is sufficiently high.

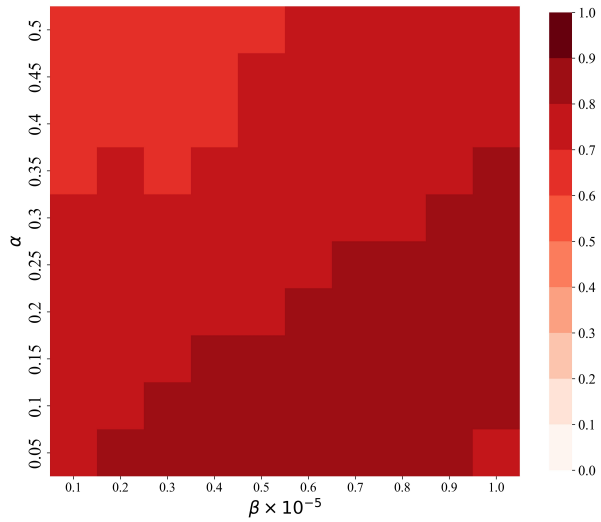



Figure 11: Expected Profits under Sym-Rigid across Parameter Space with $\delta = 0.96$

## 5.3 Observability of Demand Shocks

### 5.3.1 Symmetric Information

Regarding the observability of demand shocks, a particularly interesting question is whether agents perform differently when they can observe current market fluctuations or not. Specifically, it remains unclear whether such observability could potentially benefit or harm agents.

The impact of knowing demand shocks on profits is theoretically ambiguous. On the one hand, knowing demand shocks provides agents with stronger incentives to undercut during periods of the positive demand shocks, driving them to deviate at H. On the other hand, unobserved demand shocks introduce stochasticity into the environment, potentially impeding agents' learning by obscuring the true state-action-reward relationships.

The simulation results show that autonomous collusion and supra-competitive profits remain feasible under unobserved demand shocks, echoing the finding in Calvano et al. (2020).[41] Under this scenario, where the agents cannot observe the demand shocks, Sym-1Node (i.e., both agents charge the same price) remains the predominant pricing pattern for $\delta = 0.96$ and 0.76.

Table 6 lists the expected profits of the predominant pricing patterns under observed and unobserved demand shocks.[42] For both $\delta = 0.96$ and 0.76, agents earn higher profits when the demand shocks are unobserved. However, the difference is small, with agents under observed demand shocks earning just 1.4% and 6.9% less, respectively. These findings suggest that observing the demand shocks does not substantially impact agents' profits.

[41] One reasonable explanation is that if the distribution of demand states is known, agents can simply use the expected value as the demand state for each period. Moreover, even though the distribution may be unknown, perfect monitoring enables agents to effectively detect and punish deviating behaviors, thus making tacit collusion still possible.

[42] Recall that under observed demand shocks, the predominant pricing patterns are Sym-Rigid at $\delta = 0.96$ and Counter-Cycle at $\delta = 0.76$.

| | Observed | Unobserved |
|---|---|---|
| Observed | 6.92, 6.92 | 6.63, 6.63 |
| Unobserved | 6.63, 6.63 | 7.02, 7.02 |

(a) $\delta = 0.96$

| | Observed | Unobserved |
|---|---|---|
| Observed | 3.38, 3.38 | 1.92, 1.92 |
| Unobserved | 1.92, 1.92 | 3.63, 3.63 |

(b) $\delta = 0.76$

Table 6: Payoff Matrices

### 5.3.2 Asymmetric Information

In reality, different firms may adopt different pricing algorithms. This raises a new question: can heterogeneous algorithms still coordinate effectively to achieve collusive outcomes? Algorithm asymmetry can manifest in many ways.[43] Building upon the preceding analysis, I examine this question by investigating a scenario where one agent can observe the demand shocks while the other cannot.

The simulation results show that the predominant pricing pattern at high $\delta$ is consistently Sym-Rigid for the agent who can observe the demand shocks (and Sym-1Node for the agent who cannot). This is not surprising: with sufficiently high $\delta$, the informed agent does not have a strong incentive to deviate at H. Furthermore, due to perfect monitoring, the uninformed agent can easily detect and punish any deviating behavior. Thus, tacit collusion is still feasible.

In contrast, at $\delta = 0.76$, the predominant pricing pattern surprisingly remains Sym-Rigid (and Sym-1Node), rather than shifting to Counter-Cycle. This finding demonstrates that having only one agent informed about the current demand shock is not sufficient to enable countercyclical pricing. The absence of countercyclical pricing under asymmetric information can be explained as follows: countercyclical pricing emerges when agents charge different prices across different demand states, which necessitates that both agents be able to observe the demand shocks. However, under asymmetric information, the uninformed agent interprets the different prices charged by the informed agent as deviations, rather than as coordination across demand states. This misinterpretation leads the uninformed

[43] For instance, Brown and MacKay (2023) consider a model of price competition in which firms employ different pricing algorithms, with the asymmetry stemming from the frequency of price updates.

agent to retaliate by setting low prices. Consequently, charging different prices across demand states becomes extremely difficult, and the informed agent is compelled to adopt rigid pricing, charging the same prices across both demand states.

Table 6 also presents the expected profits under asymmetric information. Compared to the case where both agents can observe demand shocks, agents under asymmetric information earn 4.2% less when $\delta = 0.96$. However, when $\delta = 0.76$, agents earn profits of only 1.92, resulting in a substantial profit decrease of 43.2%. These results demonstrate that symmetry of information (whether demand shocks are observed or unobserved by both agents) is necessary for agents to achieve supra-competitive profits, particularly when agents discount future payoffs more heavily.

### 5.3.3 Discussion

The previous analysis shows that the algorithms earn higher profits under unobserved demand shocks than observed ones. However, this finding does not imply that predicting future demand is not important.

First, the adoption of different types of algorithms relates to the problem of equilibrium selection. Table 6 shows that there are two pure strategy Nash equilibria: (Observed, Observed) and (Unobserved, Unobserved). Although (Observed, Observed) is Pareto dominant, the choice of algorithm depends on each firm's beliefs. If the common demand shocks are typically easy to observe, then firms are more likely to adopt algorithms that incorporate observed demand shocks.

Second, the standard Bertrand competition model assumes firms can instantly adjust their production capacity at no cost to serve the entire market. While this assumption simplifies theoretical analysis, adjusting capacity and managing inventory often involves time lags and storage costs in reality. This underscores the potential value of predicting demand shocks when production capacity cannot be adjusted quickly. The question of how algorithms that observe demand shocks perform in this more realistic setting remains open

for future research.[44]

# 6 Conclusion

This paper examines algorithmic collusion in a more complex setting by introducing demand shocks into the previously static economic environment and allowing agents to observe the current demand state. This makes the economic environment more realistic while increasing coordination difficulty, as agents face unbalanced deviating incentives across demand states.

The simulation results show that when $\delta$ is sufficiently high, the symmetric rigid pricing pattern becomes predominant. Under this pricing pattern, agents consistently learn to overcome their incentives to deviate during the positive demand shocks and maintain tacit collusion. The resulting collusive profits remain at a very high level and are sustained by an effective reward-punishment scheme, demonstrating the robustness of algorithmic collusion in this more complex economic environment.

When $\delta$ is at a medium level, countercyclical pricing becomes predominant and performs better than rigid pricing. This shows that Q-learning algorithms can learn to charge lower prices during the positive demand shocks to decrease deviating incentives while maintaining relatively high prices during the negative demand shocks. The observed countercyclical pricing pattern aligns with the prediction of Rotemberg and Saloner (1986), demonstrating its applicability to algorithmic pricing and showing the strong pricing adaptability of Q-learning algorithms.

In this complex economic environment, algorithmic collusion can occur even without the use of nonpublic competitor data, posing a significant challenge to the regulation of algorithmic pricing. Furthermore, a particularly difficult issue arises with countercyclical pricing: while constant high prices across varying demand conditions can easily raise suspicions of

[44]For research on other realistic settings, Friedrich et al. (2024) examines algorithmic collusion in episodic markets with inventory constraints.

collusion, detecting potential pricing coordination becomes much more challenging when prices are more competitive during periods of positive demand shocks. More broadly, how can regulators reasonably suspect collusion when algorithmic pricing appears to adapt to fluctuating market conditions? This challenge highlights the difficulty of distinguishing between genuinely competitive behavior and potential collusion. Therefore, more sophisticated detection methods need to be developed.

## Acknowledgement

Computation reported in this work was carried out on the Unity Cluster of the College of Arts and Sciences at the Ohio State University. The computational resources and support provided are gratefully acknowledged.

# References


[1] Abreu, D. (1986). Extremal equilibria of oligopolistic supergames. *Journal of Economic Theory 39*(1), 191–225.

[2] Assad, S., E. Calvano, G. Calzolari, R. Clark, V. Denicolò, D. Ershov, J. Johnson, S. Pastorello, A. Rhodes, L. Xu, et al. (2021). Autonomous algorithmic collusion: Economic research and policy implications. *Oxford Review of Economic Policy 37*(3), 459–478.

[3] Assad, S., R. Clark, D. Ershov, and L. Xu (2024). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy 132*(3), 000–000.

[4] Athey, S., K. Bagwell, and C. Sanchirico (2004). Collusion and price rigidity. *The Review of Economic Studies 71*(2), 317–349.

[5] Ballestero, G. (2021). Collusion and artificial intelligence: A computational experiment with sequential pricing algorithms under stochastic costs.

[6] Banchio, M. and A. Skrzypacz (2022). Artificial intelligence and auction design. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 30–31.

[7] Barlo, M., G. Carmona, and H. Sabourian (2009). Repeated games with one-memory. *Journal of Economic Theory 144*(1), 312–336.

[8] Barlo, M., G. Carmona, and H. Sabourian (2016). Bounded memory folk theorem. *Journal of economic theory 163*, 728–774.

[9] Beneke, F. and M.-O. Mackenrodt (2021). Remedies for algorithmic tacit collusion. *Journal of Antitrust Enforcement 9*(1), 152–176.

[10] Brero, G., E. Mibuari, N. Lepore, and D. C. Parkes (2022). Learning to mitigate ai collusion on economic platforms. *Advances in Neural Information Processing Systems 35*, 37892–37904.

[11] Brown, Z. Y. and A. MacKay (2023). Competition in pricing algorithms. *American Economic Journal: Microeconomics 15*(2), 109–156.

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

[13] Calvano, E., G. Calzolari, V. Denicoló, and S. Pastorello (2021). Algorithmic collusion with imperfect monitoring. *International journal of industrial organization 79*, 102712.

[14] Cho, I. and N. Williams (2024). Collusive outcomes without collusion. *arXiv preprint arXiv:2403.07177*.

[15] Cowgill, B. and C. E. Tucker (2020). Algorithmic fairness and economics. *Columbia Business School Research Paper*.

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

[17] Fish, S., Y. A. Gonczarowski, and R. I. Shorrer (2024). Algorithmic collusion by large language models. *arXiv preprint arXiv:2404.00806*.

[18] Friedrich, P., B. PĂĄsztor, and G. Ramponi (2024). Learning collusion in episodic, inventory-constrained markets. *arXiv preprint arXiv:2410.18871*.

[19] Green, E. J. and R. H. Porter (1984). Noncooperative collusion under imperfect price information. *Econometrica: Journal of the Econometric Society*, 87–100.

[20] Haltiwanger, J. and J. E. Harrington Jr (1991). The impact of cyclical demand movements on collusive behavior. *The RAND Journal of Economics*, 89–106.

[21] Hanazono, M. and H. Yang (2007). Collusion, fluctuating demand, and price rigidity. *International Economic Review 48*(2), 483–515.

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

[23] Jabbari, S., M. Joseph, M. Kearns, J. Morgenstern, and A. Roth (2017). Fairness in reinforcement learning. In *International conference on machine learning*, pp. 1617–1626. PMLR.

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

[25] Kandori, M. (1991). Correlated demand shocks and price wars during booms. *The Review of Economic Studies 58*(1), 171–180.

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

[27] Kolumbus, Y. and N. Nisan (2022). Auctions between regret-minimizing agents. In *Proceedings of the ACM Web Conference 2022*, pp. 100–111.

[28] Liang, A., J. Lu, X. Mu, and K. Okumura (2021). Algorithm design: A fairness-accuracy frontier. *arXiv preprint arXiv:2112.09975*.

[29] Maskin, E. and J. Tirole (1988). A theory of dynamic oligopoly, ii: Price competition, kinked demand curves, and edgeworth cycles. *Econometrica: Journal of the Econometric Society*, 571–599.

[30] Miklós-Thal, J. and C. Tucker (2019). Collusion by algorithm: Does better demand prediction facilitate coordination between sellers? *Management Science 65*(4), 1552–1561.

[31] Nuutila, E. and E. Soisalon-Soininen (1994). On finding the strongly connected components in a directed graph. *Information Processing Letters 49*(1), 9–14.

[32] O'Connor, J. and N. E. Wilson (2021). Reduced demand uncertainty and the sustainability of collusion: How ai could affect competition. *Information Economics and Policy 54*, 100882.

[33] Rambachan, A., J. Kleinberg, J. Ludwig, and S. Mullainathan (2020). An economic perspective on algorithmic fairness. In *AEA Papers and Proceedings*, Volume 110, pp. 91–95. American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203.

[34] Raza, M. Q. and A. Khosravi (2015). A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews 50*, 1352–1372.

[35] Rotemberg, J. J. and G. Saloner (1986). A supergame-theoretic model of price wars during booms. *The American Economic Review 76*(3), 390–407.

[36] Sargent, T. J. and J. Stachurski (2024). *Economic networks: Theory and computation*, Volume 53. Cambridge University Press.

[37] Seyedan, M. and F. Mafakheri (2020). Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data 7*(1), 53.

[38] Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.

[39] Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing 1*(2), 146–160.

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

[41] Watkins, C. J. C. H. (1989). Learning from delayed rewards.

[42] Xu, Z. and W. Zhao (2024). On mechanism underlying algorithmic collusion. *arXiv preprint arXiv:2409.01147*.

# Appendix

## A Tables

Table A.1: Deviating Agent's Frequencies of Profitable Deviations at $\delta = 0.96$

| Stable | | Deviation occurring at L | | | | | | | | | Stable | | Deviation occurring at H | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| price | Proportion | 0.5[a] | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5 | price | Proportion | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5 |
| *Panel A: Under Demand Shocks — Sym-Rigid* | | | | | | | | | | | | | | | | | | | | | |
| 1.5[b] | 0.008 | 0.13 | 0.26 | | | | | | | | 1.5 | 0.008 | 0.11 | 0.33 | | | | | | | |
| 2 | 0.272 | 0.07 | 0.13 | 0.20 | | | | | | | 2 | 0.272 | 0.04 | 0.13 | 0.36 | | | | | | |
| 2.5 | 0.346 | 0.03 | 0.10 | 0.12 | 0.16 | | | | | | 2.5 | 0.346 | 0.01 | 0.05 | 0.15 | 0.31 | | | | | |
| 3 | 0.219 | 0.03 | 0.10 | 0.11 | 0.12 | 0.15 | | | | | 3 | 0.219 | 0.00 | 0.04 | 0.11 | 0.20 | 0.29 | | | | |
| 3.5 | 0.095 | 0.03 | 0.06 | 0.09 | 0.18 | 0.13 | 0.17 | | | | 3.5 | 0.095 | 0.00 | 0.02 | 0.06 | 0.22 | 0.19 | 0.30 | | | |
| 4 | 0.043 | 0.02 | 0.08 | 0.08 | 0.17 | 0.10 | 0.16 | 0.12 | | | 4 | 0.043 | 0.00 | 0.05 | 0.05 | 0.17 | 0.16 | 0.24 | 0.24 | | |
| 4.5 | 0.014 | 0.02 | 0.04 | 0.13 | 0.11 | 0.09 | 0.17 | 0.06 | 0.07 | | 4.5 | 0.014 | 0.00 | 0.00 | 0.07 | 0.09 | 0.11 | 0.25 | 0.15 | 0.21 | |
| 5 | 0.002 | 0.00 | 0.15 | 0.33 | 0.13 | 0.13 | 0.12 | 0.13 | 0.13 | 0.00 | 5 | 0.002 | 0.00 | 0.10 | 0.00 | 0.12 | 0.13 | 0.14 | 0.13 | 0.13 | 0.10 |
| *Panel B: At Single Demand State — Sym-1Node* | | | | | | | | | | | | | | | | | | | | | |
| 1.5 | 0.147 | 0.02 | 0.13 | | | | | | | | 1.5 | 0 | | | | | | | | | |
| 2 | 0.518 | 0.01 | 0.06 | 0.28 | | | | | | | 2 | 0.033 | 0.00 | 0.06 | 0.09 | | | | | | |
| 2.5 | 0.238 | 0.00 | 0.05 | 0.13 | 0.23 | | | | | | 2.5 | 0.308 | 0.01 | 0.01 | 0.09 | 0.24 | | | | | |
| 3 | 0.077 | 0.00 | 0.01 | 0.13 | 0.18 | 0.21 | | | | | 3 | 0.317 | 0.01 | 0.02 | 0.06 | 0.09 | 0.23 | | | | |
| 3.5 | 0.016 | 0.00 | 0.07 | 0.07 | 0.53 | 0.33 | 0.40 | | | | 3.5 | 0.202 | 0.01 | 0.01 | 0.04 | 0.05 | 0.14 | 0.17 | | | |
| 4 | 0.003 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.33 | 0.33 | | | 4 | 0.078 | 0.00 | 0.00 | 0.01 | 0.01 | 0.05 | 0.12 | 0.17 | | |
| 4.5 | 0 | | | | | | | | | | 4.5 | 0.042 | 0.00 | 0.00 | 0.05 | 0.00 | 0.02 | 0.17 | 0.10 | 0.15 | |
| 5 | 0 | | | | | | | | | | 5 | 0.020 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 | 0.05 | 0.16 | 0.11 | 0.16 |

Notes: a. The price (row) denotes the undercutting price.
b. The price (column) denotes the stable price before deviation happens.

Table A.2: Deviating Agent's Profit Ratios at $\delta = 0.96$ (Deviation Path Relative to Non-deviation Path)

| Stable | | Deviation occurring at L | | | | | | | | | Stable | | Deviation occurring at H | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| price | Proportion | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5 | price | Proportion | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5 |
| *Panel A: Under Demand Shocks — Sym-Rigid* | | | | | | | | | | | | | | | | | | | | | |
| 1.5 | 0.008 | 0.79 | 0.88 | | | | | | | | 1.5 | 0.008 | 0.77 | 0.94 | | | | | | | |
| 2 | 0.272 | 0.71 | 0.78 | 0.85 | | | | | | | 2 | 0.272 | 0.68 | 0.81 | 0.93 | | | | | | |
| 2.5 | 0.346 | 0.64 | 0.71 | 0.77 | 0.81 | | | | | | 2.5 | 0.346 | 0.61 | 0.72 | 0.82 | 0.91 | | | | | |
| 3 | 0.219 | 0.61 | 0.68 | 0.73 | 0.77 | 0.79 | | | | | 3 | 0.219 | 0.57 | 0.68 | 0.76 | 0.84 | 0.91 | | | | |
| 3.5 | 0.095 | 0.58 | 0.65 | 0.68 | 0.78 | 0.74 | 0.77 | | | | 3.5 | 0.095 | 0.53 | 0.64 | 0.70 | 0.82 | 0.84 | 0.91 | | | |
| 4 | 0.043 | 0.61 | 0.67 | 0.69 | 0.76 | 0.73 | 0.78 | 0.73 | | | 4 | 0.043 | 0.54 | 0.64 | 0.69 | 0.79 | 0.80 | 0.88 | 0.89 | | |
| 4.5 | 0.014 | 0.49 | 0.56 | 0.65 | 0.67 | 0.67 | 0.78 | 0.67 | 0.66 | | 4.5 | 0.014 | 0.44 | 0.52 | 0.63 | 0.69 | 0.73 | 0.86 | 0.81 | 0.85 | |
| 5 | 0.002 | 0.46 | 0.61 | 0.74 | 0.61 | 0.64 | 0.64 | 0.59 | 0.52 | 0.48 | 5 | 0.002 | 0.41 | 0.57 | 0.61 | 0.61 | 0.68 | 0.71 | 0.70 | 0.72 | 0.72 |
| *Panel B: At Single Demand State — Sym-1Node* | | | | | | | | | | | | | | | | | | | | | |
| 1.5 | 0.147 | 0.69 | 0.86 | | | | | | | | 1.5 | 0 | | | | | | | | | |
| 2 | 0.518 | 0.66 | 0.79 | 0.90 | | | | | | | 2 | 0.033 | 0.61 | 0.75 | 0.83 | | | | | | |
| 2.5 | 0.238 | 0.61 | 0.75 | 0.83 | 0.90 | | | | | | 2.5 | 0.308 | 0.60 | 0.71 | 0.80 | 0.89 | | | | | |
| 3 | 0.077 | 0.59 | 0.73 | 0.82 | 0.87 | 0.91 | | | | | 3 | 0.317 | 0.56 | 0.68 | 0.76 | 0.82 | 0.88 | | | | |
| 3.5 | 0.016 | 0.68 | 0.77 | 0.81 | 0.96 | 0.95 | 0.97 | | | | 3.5 | 0.202 | 0.54 | 0.63 | 0.70 | 0.75 | 0.82 | 0.85 | | | |
| 4 | 0.003 | 0.56 | 0.76 | 0.67 | 0.72 | 0.83 | 0.78 | 1.08 | | | 4 | 0.078 | 0.52 | 0.65 | 0.66 | 0.73 | 0.79 | 0.85 | 0.88 | | |
| 4.5 | 0 | | | | | | | | | | 4.5 | 0.042 | 0.52 | 0.61 | 0.66 | 0.70 | 0.75 | 0.82 | 0.80 | 0.85 | |
| 5 | 0 | | | | | | | | | | 5 | 0.020 | 0.52 | 0.63 | 0.68 | 0.74 | 0.78 | 0.80 | 0.88 | 0.87 | 0.88 |

Notes: The settings are identical to those in Table A.1 in the Appendix.

# B Figures

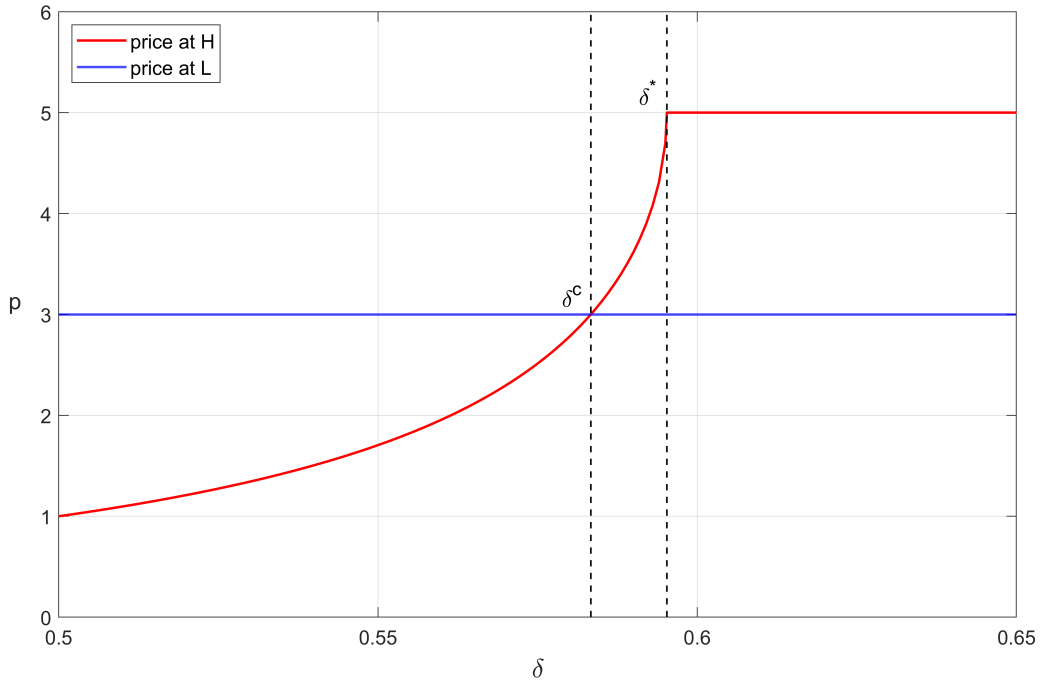
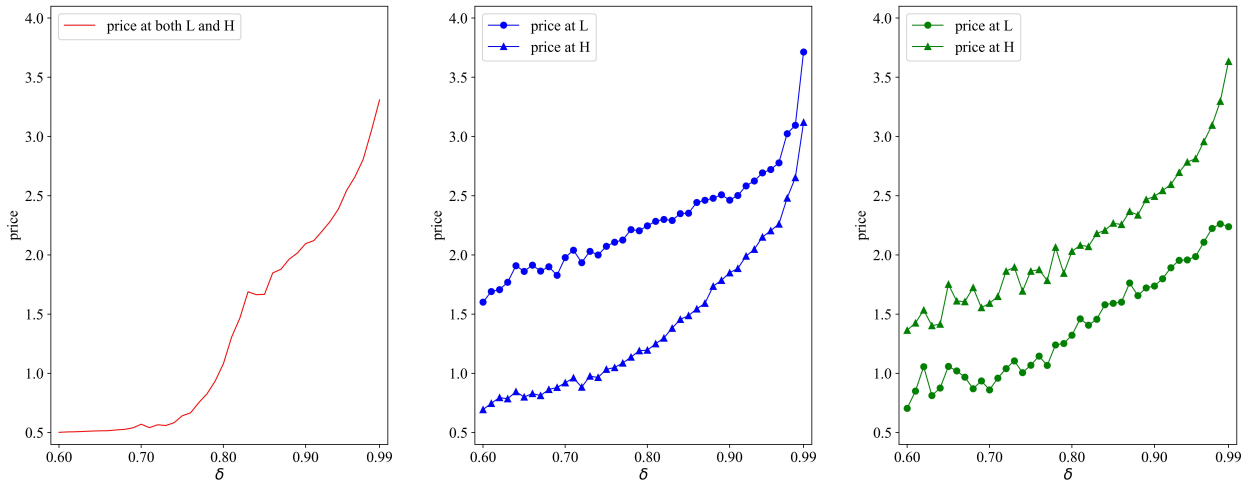
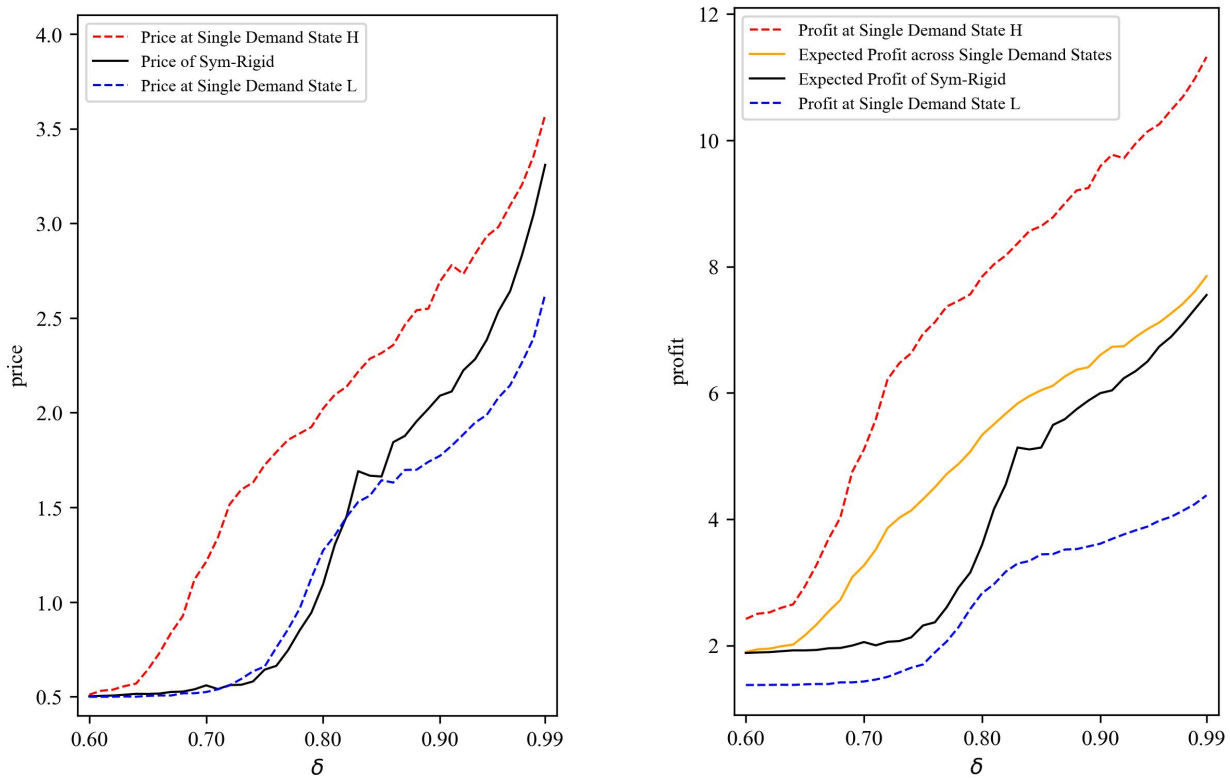



Figure A.1: Predicted Prices for Both Demand States across $\delta$




(a) Sym-Rigid (b) Counter-Cycle (c) Pro-Cycle

Figure A.2: Price Dynamics of Three Pricing Patterns

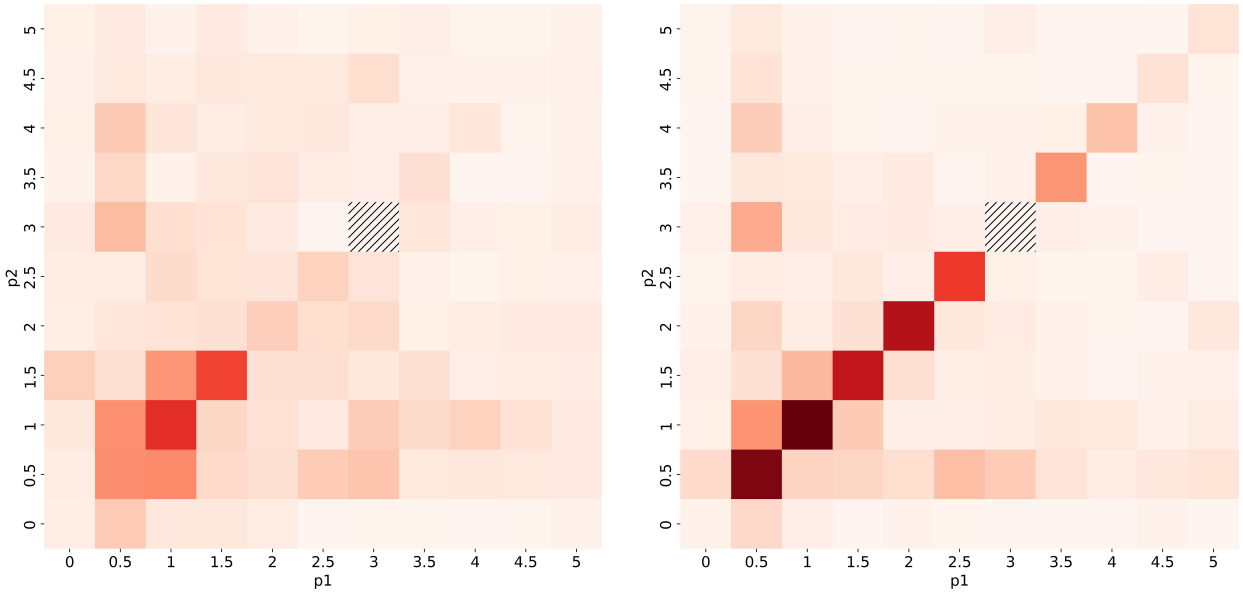



(a) Price

(b) Profit

Figure A.3: Price and Profit Dynamics of Sym-Rigid and Sym-1Node

Notes: The black line represents the average prices and profits of Sym-Rigid, while the dashed lines, each in a different color, indicate prices and profits at the single demand states (Sym-1Node). In Panel B, the yellow line represents the average profit across the two single demand states.




(a) Price Pairs at L

(b) Price Pairs at H

Figure A.4: In-Degree Centrality of Price Pairs on the Deviation Path (Deviation Initiating at L)

Notes: The square depicted in lines marks the pre-deviation price of 3. A deviation initiates when agent 1 undercuts by the minimum price unit after observing a negative demand shock. The color gradient indicates the level of in-degree centrality.

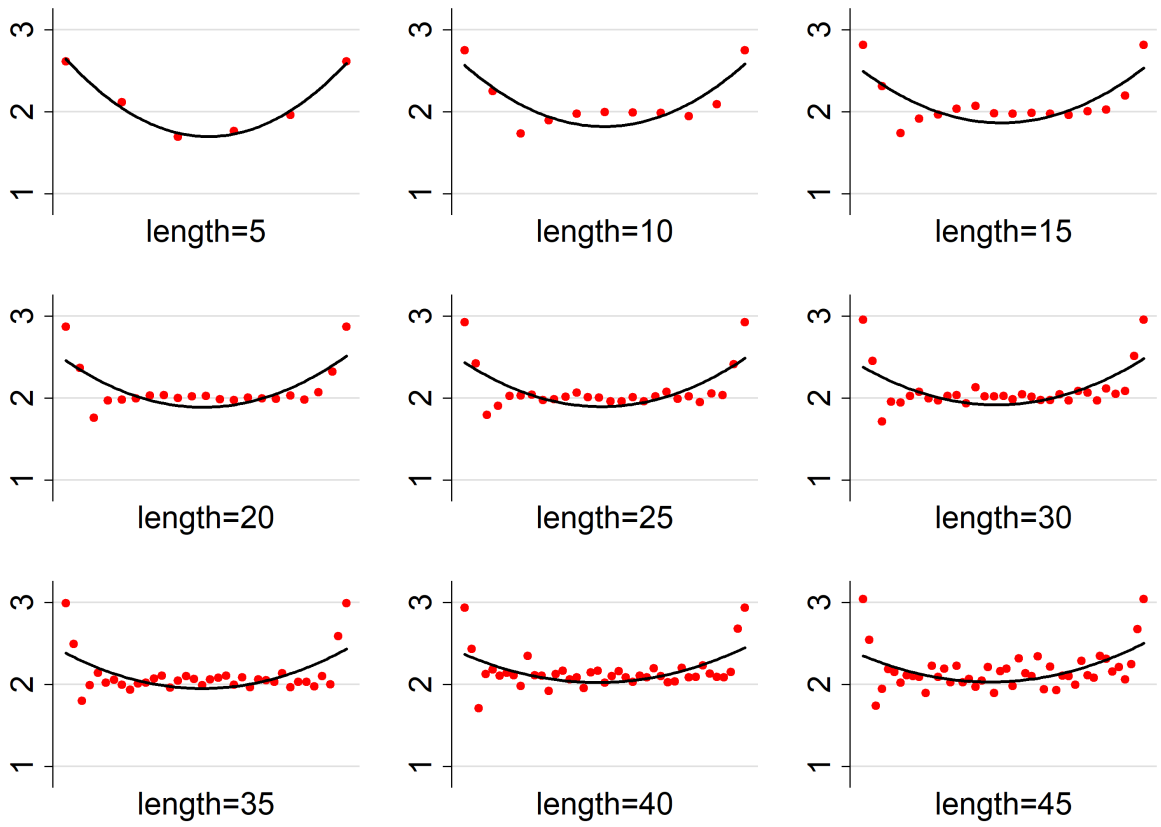
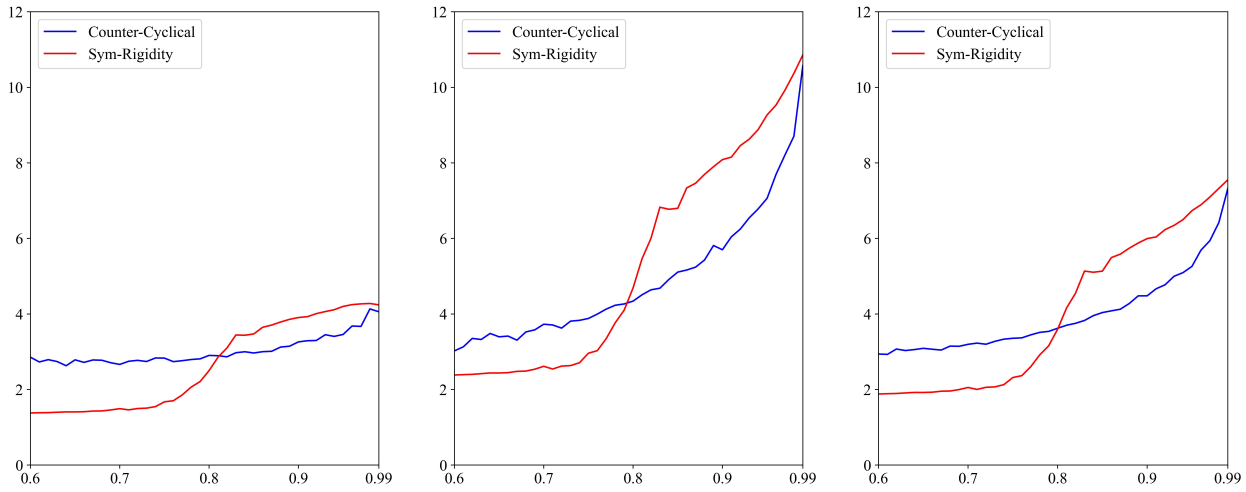



Figure A.5: Deviating Agent's Price Dynamics under Various Lengths of Deviation Paths

Notes: The black line denotes the fitted prices from the regression model (regressing price on period and other control variables), while the red nodes represent the average price across demand states and sessions in each period.




(a) Profit at L (b) Profit at H (c) Expected Profit

Figure A.6: Profit Comparison: Counter-Cycle vs. Sym-Rigid

# C Initial Q-matrix

The initial Q-matrix $\mathbf{Q}_{i0}$ in the baseline analysis is calculated as follows. Given that the opponent randomly chooses a price at period $t = 0$, the expected period payoff for agent $i$ who sets price $p_i$ at demand state $\theta$ is

$$\bar{\pi}_i(p_i, \theta) = \frac{\sum_{p_{-i} \in A} \pi_i(p_i, p_{-i}, \theta)}{|A|}$$

Correspondingly, the initial Q-value at entry $(s_\theta, p)$, where $s_\theta$ denotes any state with demand shock $\theta$, is

$$Q_{i0}(\theta, p_i) = \bar{\pi}_i(p_i, \theta) + \delta \frac{\sum_{\theta_j \in \Theta} Q_{i0}(\theta_j, p_i)}{|\Theta|}$$

where $Q_{i0}(s_\theta, p_i)$ is written as $Q_{i0}(\theta, p_i)$ for simplicity.

Then the initial Q-values at $p_i$ for each demand state can be solved through the linear

equation system

$$\begin{aligned}
Q_{i0}(\theta_1, p_i) &= \bar{\pi}_i(p_i, \theta_1) + \delta \frac{\sum_{\theta_j \in \Theta} Q_{i0}(\theta_j, p_i)}{|\Theta|} \\
Q_{i0}(\theta_2, p_i) &= \bar{\pi}_i(p_i, \theta_2) + \delta \frac{\sum_{\theta_j \in \Theta} Q_{i0}(\theta_j, p_i)}{|\Theta|} \\
\vdots \qquad & \qquad \vdots \qquad \qquad \vdots \\
Q_{i0}(\theta_{|\Theta|}, p_i) &= \bar{\pi}_i(p_i, \theta_{|\Theta|}) + \delta \frac{\sum_{\theta_j \in \Theta} Q_{i0}(\theta_j, p_i)}{|\Theta|}
\end{aligned}$$

# D Theoretical Prediction

Figure A.1 illustrates the theoretical prediction of prices under different values of $\delta$. For the negative demand shock (L), agents can charge the monopoly price of 3 as long as $\delta \geq 0.5$. For the positive demand shock (H), there exists two critical values of $\delta$. When $\delta > \delta^* = \frac{25}{42} \approx 0.595$, agents have no incentive to deviate from charging the monopoly price of 5 at H, thus sustaining the fully collusive outcome.[45] When $\delta < \delta^*$, the fully collusive outcome cannot be sustained, leading agents to maintain partial collusion by lowering prices at H.

When $\delta$ falls below the cutoff $\delta^c = \frac{7}{12} \approx 0.583$, agents further lower prices, resulting in countercyclical pricing. Notably, despite lower prices at H, higher profits are still earned at H when $\delta \in (0.5, \delta^c]$.

# E Average Long-run Prices

The transitions on $G_c$ satisfy the Markov property, where each node has exactly two direct successors with equal transition probability of 0.5. Therefore, this defines a finite Markov process. Since $G_c$ is strongly connected, the existence and uniqueness of its stationary distribution are guaranteed. The stationary distribution $\psi^*$ is defined by $\psi^* P = \psi^*$, where $P$ is the stochastic adjacency matrix (Markov matrix).[46]

Let $n$ denote the number of nodes in $G_c$. Let $\mathbb{1}_n$ be the $1 \times n$ row vector $(1, \ldots, 1)$ and $\mathbb{1}_{n \times n}$ be the $n \times n$ matrix of ones. With $I$ as the identity matrix and $P$ as the stochastic adjacency

[45] Note that agents always have a stronger incentive to deviate at H, i.e., when observing the positive demand shocks.

[46] The theorem from (36) states that every finite Markov model has at least one stationary distribution $\psi^*$. If the digraph is strongly connected, then $\psi^*$ is unique and everywhere positive.

matrix of $G_c$, the unique stationary distribution $\psi^*$ can be solved by

$$\mathbb{1}_n = \psi^*(I - P + \mathbb{1}_{n\times n}) \tag{6}$$

Using the stationary distribution $\psi^*$, the average long-run price for each demand state can be calculated. To illustrate, consider the stochastic adjacency matrix $P$ in Figure 1b:

$$\begin{array}{c} \\ L-(2,2) \\ H-(2,2) \\ H-(4,4) \end{array} \begin{array}{c} \begin{array}{ccc} L-(2,2) & H-(2,2) & H-(4,4) \end{array} \\ \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{pmatrix} \end{array}$$

Solving equation (6) yields the stationary distribution $\psi^* = (\frac{1}{2}, \frac{1}{6}, \frac{1}{3})$, where each element represents the steady-state probability of the corresponding node. The average long-run price pairs at $L$ and $H$ are $(2,2)$ and $(\frac{10}{3}, \frac{10}{3})$, respectively.

## F Algorithm for Deviation Tests

Algorithm 2 illustrates one simulation in the deviation test. $\Pi_i^*$ and $\Pi_i^D$ denote agent $i$'s accumulated discounted profits on the non-deviation and deviation paths, respectively. The corresponding period profits are denoted by $\pi_{it}^*$ and $\pi_{it}^D$.

**Algorithm 2** One Simulation for Deviation Test

***First step: Initialization***

$\theta_0$ is realized

$(\Pi_1^*, \Pi_2^*) \leftarrow (\pi_{11}^*, \pi_{21}^*)$ ▷ *on the non-deviation path*

Let one agent undercut

$(\Pi_1^D, \Pi_2^D) \leftarrow (\pi_{11}^D, \pi_{21}^D)$ ▷ *on the deviation path*

***Second step: Loop***

**while** Price dynamics do *not* return to $G_c$ **do**

    $\theta_t$ is realized

    $(\Pi_1^*, \Pi_2^*) \leftarrow (\Pi_1^* + \delta^t \pi_{1t}^*, \Pi_2^* + \delta^t \pi_{2t}^*)$

    $(\Pi_1^D, \Pi_2^D) \leftarrow (\Pi_1^D + \delta^t \pi_{1t}^D, \Pi_2^D + \delta^t \pi_{2t}^D)$

    $t \leftarrow t + 1$