Learning Collusion in Episodic, Inventory-Constrained Markets

Anonymous Author(s)
Affiliation
Address
email

Abstract

Pricing algorithms have demonstrated the capability to learn tacit collusion that is largely unaddressed by current regulations. Their increasing use in markets, including oligopolistic industries with a history of collusion, calls for closer examination by competition authorities. In this paper, we extend the study of tacit collusion in learning algorithms from basic pricing games to more complex markets characterized by perishable goods with fixed supply and sell-by dates, such as airline tickets, perishables, and hotel rooms. We formalize collusion within this framework and introduce a metric based on price levels under both the competitive (Nash) equilibrium and collusive (monopolistic) optimum. Since no analytical expressions for these price levels exist, we propose an efficient computational approach to derive them. Through experiments, we demonstrate that deep reinforcement learning agents can learn to collude in this more complex domain. Additionally, we analyze the underlying mechanisms and structures of the collusive strategies these agents adopt.

1 Introduction

Algorithms are increasingly replacing humans in pricing decisions, offering improved revenue management and handling of complex dynamics in large-scale markets such as retail and airline ticketing. These algorithms, whether programmed or self-learning, can engage in tacit collusion charging *supra-competitive* prices (i.e., above the competitive level) or limiting production without explicit agreements. For example, algorithmic pricing in Germany led to a 38% increase in fuel retailer margins after adoption (Assad et al., 2024). Our study is primarily motivated by airline revenue management (ARM), a market with $800 billion in annual revenue and thin profit margins. Airlines have already been under regulatory scrutiny (European Union, 2019) due to evidence of tacit collusion even before the introduction of algorithmic pricing (Borenstein & Rose, 1994) but the current trend of moving towards algorithmic pricing (Koenigsberg, Muller, & Vilcassim, 2004; Razzaghi et al., 2022) could lead to further cases.

Tacit collusion is maintained without explicit communication or agreement between sellers, therefore it eludes detection and often falls outside the scope of current competition laws. These concerns and potential negative effects on social welfare have been recognized both by regulators (Ohlhausen, 2017; Bundeskartellamt & Autorité de la Concurrence, 2019; Directorate-General for Competition (European Commission) et al., 2019) and scholars (Harrington, 2018; Beneke & Mackenrodt, 2021; Brero et al., 2022). To develop comprehensive legislation on algorithmic pricing, a thorough understanding of the factors that influence the emergence of collusive strategies is required under assumptions that align with real markets (Calvano et al., 2020b).

Previous research has already shown that *reinforcement learning (RL)* algorithms can engage in tacit collusion in pricing games with infinite time-horizon (Asker, Fershtman, & Pakes, 2022; Calvano

Link to github and author correspondence will be added here after acceptance.

et al., 2020a; Musolff, 2022; Klein, 2021). However, most markets follow some form of periodicity, e.g., seasonality in retail or fiscal years for public companies, which breaks the continuity of the interactions between sellers. In the markets of perishable goods, hotels, or tickets, the markets only persist until the given sell-by dates and sellers are aware of the finite nature of competition. Importantly, in the previously investigated infinite time-horizon settings the collusive equilibrium is maintained via punishment strategies, e.g., grim-trigger, but it is not an equilibrium in the finite-horizon case. This is because these strategies are only credible if sufficient time remains for the punishment to offset short-term gains from deviating from collusion. In the finite-horizon setting, such punishments become unmaintainable as the sell-by date approaches. However, RL algorithms show the potential to learn collusion through their memory over several episodes interacting against the same opponents. Additionally, in finite time-horizon markets supplies are often predetermined and limited, therefore, pricing strategies have to consider additional constraints and anticipate future demand to avoid expiring inventory while maximizing total profit. Both aspects are crucial in many real-world markets. For example, airlines selling tickets between two cities on a certain day have to fill their planes' capacity before departure. However, selling tickets too quickly could lead to a missed opportunity to sell tickets closer to departure time to less price-sensitive consumers, while selling tickets too slowly could result in empty seats. The added complexity of finite time horizon and inventory constraints results in more complex strategies and interactions between pricing algorithms; therefore, previous results do not immediately hold and further investigation is necessary to develop comprehensive collusion mitigation approaches.

In this work, we aim to contribute to these efforts by extending the analysis of tacit collusion between pricing algorithms to *episodic markets with inventory constraints*.

In particular, in Section 2, we give an overview of related literature. In Section 3, we define the episodic, finite-horizon pricing problem with inventory constraints as a Markov game, inspired by Airline Revenue Management (ARM), and formalize both competitive (Nash) and collusive (monopolistic) equilibrium strategies. Building on these, we define a measure that quantifies collusion in an observed episode. Notably, our definitions are on the space of pricing strategies instead of price levels at a certain point in time which is the standard in the infinite-horizon setting. This is a significant change in the analysis and a challenge in episodic markets compared to the infinite-horizon case. In Section 4, we discuss how our model's finite time horizon and inventory constraints change the dynamics of collusion compared to previous work. Reward-punishment schemes cannot extend past the end of the episode, making collusion theoretically impossible (with a backward induction argument), but practically achievable (with imperfect learning agents and long enough episodes). In Section 5, we demonstrate efficient computation of the competitive Nash Equilibrium, a challenging task on its own. We show that two common deep RL algorithms, Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Deep Q-Networks (DQN) (Mnih et al., 2015), learn to collude in our model in two distinct ways that align with the intuition provided in Section 4. We analyze the learned strategies, finding that agents collude while being aware of the competitive best response, and maintain collusion with a reward-punishment scheme. We show that collusion is robust to changes in agent hyperparameters, unless learning targets are made intentionally unstable, in which case agents converge to a competitive best response strategy. In Section 6, we conclude and discuss future research directions.

2 Related work

Our work is related to a line of research into competitive and collusive dynamics that emerge between reinforcement learning algorithmic pricing agents in economic games. We refer to Abada et al. (2024) for an excellent survey on this topic, and to Appendix C for a more detailed literature review.

Recent research most relevant to us focuses on the Bertrand oligopoly, where agents compete by setting prices and using Q-learning. The main line of research uses Bertrand competition with an infinite time horizon (Calvano et al., 2020a), with follow-up work using DQN (Hettich, 2021), varying the demand model (Asker, Fershtman, & Pakes, 2022), modeling sequential rather than simultaneous agent decisions (Klein, 2021), or an episodic setting with contexts (Eschenbaum, Mellgren, & Zahn, 2022). Findings reveal frequent, though not universal, collusion emergence, often explained by environmental *non-stationarity* preventing theoretical convergence guarantees. Agents consistently learn to charge supra-competitive prices, punishing deviating agents through 'price wars' before reverting to collusion. The robustness of collusion emergence to factors like agent number, market power asymmetry, and demand model changes underscores the potential risks posed by AI in pricing.

Which factors support and impede the emergence of learned collusion remain debated. Some (Waltman & Kaymak, 2008; Abada & Lambin, 2023) argue collusion results from agents ‘locking in’ on supra-competitive prices early on due to insufficiently exploring the strategy space, suggesting a dependence on the choice of hyperparameters. Most studies identifying collusion used Q-learning, with others showing competitive behavior, raising questions about algorithm specificity (Sanchez-Cartas & Katsamakas, 2022). However, recent work (Koirala & Laine, 2024; Deng, Schiffer, & Bichler, 2024) using PPO in ridesharing markets and infinite Bertrand competition respectively, suggests otherwise. We expand on these findings in a more realistic episodic, finite horizon market with inventory constraints using deep RL algorithms (PPO and DQN), to manage our model's larger state spaces and dynamic environments.

3 Problem statement

We introduce a multi-agent market model for inventory-constrained goods with a sell-by date, such as perishable items, hotel rooms, or tickets, using airline revenue management (ARM) as an example. We show how to model such markets as a Markov game and define a collusion metric based on the profits achieved under perfect competition and collusion.

3.1 Episodic Markov games

An *episodic Markov game* (Littman, 1994) is defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, T)$ where $\mathcal{S}$ represents the common state space shared by all agents, $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ denotes the joint action space for n agents, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the stochastic state transition function, $R_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ defines the reward received by agent i, and T specifies the episode length in discrete timesteps.

At each time step t, agents observe the current state $s_t \in \mathcal{S}$ and simultaneously choose actions following their respective time-dependent policies $\pi_{i,t} : \mathcal{S} \to \mathcal{P}(\mathcal{A}_i)$. We use π_i to denote agent i's vector of policies over time. Each agent's goal is to maximize its cumulative reward over the episode given the game's dynamics,

$$\begin{aligned} &\max_{\pi_i} \sum_{t=1}^{T} R_i(s_t, a_t) \\ \textbf{s.t.} \quad & s_{t+1} \sim P(s_t, a_t); \quad a_{j,t} \sim \pi_{j,t}(s_t). \end{aligned}$$

The main challenge in finding optimal policies in a Markov game is that agent i's optimization problem depends on the actions chosen by all other agents. In a learning context, where agents optimize their policies simultaneously, this optimization becomes non-stationary and convergence is not guaranteed. For a detailed discussion on the challenges of multi-agent reinforcement learning we refer the reader to a number of surveys Buşoniu, Babuška, and De Schutter, 2010; Yang and Wang, 2021; Gronauer and Diepold, 2022; Wong et al., 2023.

3.2 Markets as episodic Markov games

We extend the *Bertrand competition* (Bertrand, 1883) model, where agents compete to sell a common good. In its simplified one-shot setting, sellers choose prices, and consumers react, deciding which quantity to buy from each seller based on some demand function of those prices. In contrast, we model markets where goods can be sold in multiple timesteps $t = 1, \dots, T$ over a finite episode. The Markov game's action space $\mathcal{A}$ consists of the prices agents can set, and an agent's policy π_i represents their pricing strategy. Each timestep t, agents observe the state s_t and simultaneously use their policy π_i to choose an action in the form of a *price* $p_{i,t} = \pi_i(s_t)$, forming the price vector $p_t = (p_{1,t}, \dots, p_{n,t})$. In the following, we use $p_{i,t}$ for actions instead of $a_{i,t}$ to emphasize that the actions represent prices.

Additionally, we assume that each agent has a finite *capacity* $I_i \in \mathbb{N}$ of goods that they can sell throughout the episode. At each time t, each agent has a remaining *inventory* of tickets $x_{i,t} \in \{0, \dots, I_i\}$, resulting in an inventory vector $x_t = (x_{1,t}, \dots, x_{n,t})$. We define the state of the game at time t as the most recent price vector and current inventory, i.e., $s_t = (p_{t-1}, x_t)$. We motivate this definition of the state by the fact that in the non-episodic setting, most recent prices provide agents sufficient information to learn various strategies including perfect competition and collusion (Calvano et al., 2020a; Eschenbaum, Mellgren, & Zahn, 2022). However, investigating the effect of longer recall is an interesting direction for future research.

With prices chosen, a state transition from time t to $t + 1$ occurs: For each agent i, the market determines a *demand* $d_{i,t}$, the agent sells a corresponding *quantity* $q_{i,t} = \min(d_{i,t}, x_{i,t})$ bounded by their inventory, and their inventory is updated to $x_{i,t+1} = x_{i,t} - q_{i,t}$. With our choice of demand function (cf. Section 3.4), this transition to the next period's state $s_{t+1} = (p_t, x_{t+1})$ is deterministic. Finally, each agent receives their profit as a *reward* $R_{i,t} := R_i(s_t, p_t) = (p_{i,t} - c_i)q_{i,t}$, with c_i their constant *marginal cost* per good sold.

3.3 Application to airline revenue management

To motivate the episodic Markov game framework, we consider the Airline Revenue Management (ARM) problem. In ARM, agents represent airlines competing to sell a fixed number of seats on a direct flight (also called a *single-leg* flight) between two cities on the same day. The problem is naturally episodic; episodes start when the flight schedule is announced and end at departure, i.e., the sell-by date of the tickets. Furthermore, each airline is constrained by the capacity of their respective aircraft. We consider each route on each day to form a single independent market. Expanding our model to connecting (*multi-leg*) flights, several flights on the same day, cancellations, and overbooking promises interesting future work. This market is a great example with fierce competition, a history of tacit collusion (Borenstein & Rose, 1994), real-time public information on offered ticket prices and inventories via Global Distribution Systems (GDS), and early adoption of dynamic pricing algorithms (Koenigsberg, Muller, & Vilcassim, 2004)[1].

3.4 Demand model

We employ a modified *multinomial logit (MNL)* demand model, commonly used in Bertrand price competition (Calvano et al., 2020a; Eschenbaum, Mellgren, & Zahn, 2022; Deng, Schiffer, & Bichler, 2024), to simulate the probability of a customer choosing each agent's product, ensuring demand distribution among all agents rather than clustering on the best offering. The normalized *demand* for agent i's good in period t is

$$d_{i,t} = \frac{\exp\big((\alpha_i - p_{i,t})/\mu\big)}{\sum_{j \in N_t^a} \exp\big((\alpha_j - p_{j,t})/\mu\big) + \exp(\alpha_0/\mu)} \in (0,1),$$

where $N_t^a := \{j \in N \mid x_{j,t} > 0\}$, α_i is agent i's good's quality, α_0 is the quality of an outside good for vertical differentiation, and μ is the horizontal differentiation scaling parameter. The quantity demanded from agent i at time t is then defined as $q_{i,t} = \min\{\lfloor \lambda d_{i,t} \rceil, x_{i,t}\}$, scaling demand with a factor $\lambda \in \mathbb{N}$ and rounding to the nearest integer to account for the sale of goods in whole numbers. We incorporate *choice substitution*, or *demand adaptation*, by summing only over agents with available inventory N_t^a. If an agent is sold out, demand shifts to those with remaining inventory, preventing the sold-out agent's actions from affecting the demand and rewards of others.

3.5 Measuring collusion and competition

We measure the collusion of an observed episode and agent strategies on a scale from 0 (*competitive*) to 1 (*collusive*). First, we establish the two extremes in the Markov game as the competitive Nash equilibrium and the monopolistic optimum that we can later use as reference points for collusion.

Definition 3.1 (Competitive & collusive solutions)**.** A collection of agent policies $(\pi_1, \ldots, \pi_n)$ is called

- *Competitive*, or *Nash equilibrium*, if no agent i can improve their expected episode profit $\mathbb{E}_\pi[\Sigma_{t=1}^T R_{i,t}]$ by unilaterally picking a different policy given fixed opponent policies.
- *Collusive*, or *monopolistic optimum*, if it maximizes expected collective profits, $\mathbb{E}_\pi[\Sigma_{i=1}^n \Sigma_{t=1}^T R_{i,t}]$.

As we argue theoretically in Section 4 and show experimentally in Section 5.1, both admit solutions that feature constant prices across an episode, which we call p^N and p^M for the Nash and monopoly cases, respectively. In our model, the collusive prices p^M are higher than the competitive prices p^N, and the same holds for the correspondingly achieved profits R^M and R^N. At the Nash equilibrium, both unilaterally increasing or decreasing one's price reduces profits. However, if all agents jointly increase prices, the increase in margin outpaces the decrease in (MNL) demand, leading to increased profits for everyone. Building on these two solutions, we define a measure for collusion.

[1] Adoption of dynamic pricing algorithms in this industry has historically been limited to low-cost carriers, due to established carriers heavily depending on legacy systems and data-driven forecasting models. See lit. review in Appendix C.

Definition 3.2 (Collusion measure)**.** We define agent i's *episodic profit gain* as

$$\Delta_{i,e} := \frac{1}{T}\sum_{t=1}^{T}\frac{\bar{R}_{i,t} - R^{N}_{i,t}}{R^{M}_{i,t} - R^{N}_{i,t}}.$$

The *episodic collusion index* is measured as the generalized mean of the individual episodic profit gains, i.e.,

$$\Delta_e := \left(\frac{1}{n}\sum_{i=1}^{n}\Delta^{\gamma}_{i,e}\right)^{\frac{1}{\gamma}}$$

indicating a competitive or collusive outcome at 0 or 1, respectively.

The generalized mean interpolates the arithmetic mean (i.e., average) and geometric mean, which are obtained by setting $\gamma = 1$ and $\gamma = 0$ respectively. We use $\gamma = 0.5$ for our collusion index. Our reason is that the geometric mean has an advantage against the simple average used in previous studies (Calvano et al., 2020a; Eschenbaum, Mellgren, & Zahn, 2022), as it more strongly penalizes unilateral competitive defections in a collusive arrangement. However, it interprets any outcome where at least one agent achieves only competitive, or even sub-competitive profits (defining the measure via clamping negative profit gains to zero) as fully competitive, even if others prices above the competitive level and achieve considerable supra-competitive profits. The generalized mean provides a good middle ground. To better interpret negative values, we replace $\Delta^{\gamma}_{i,e}$ with $\mathrm{sgn}(\Delta_{i,e})|\Delta_{i,e}|^{\gamma}$. See Appendix E.1 for a comparison of means. Ultimately, how to aggregate the individual profit gains is a subjective question with trade-offs that depend on which outcomes one wants to differentiate the best. E.g., the following outcomes $(\Delta_{1,e}, \Delta_{2,e})$ of $(0.1, 0.1)$, $(0, 0.2)$ or $(-0.1, 0.3)$ have the same average episodic profit gain, but quite different agent behavior and implications on consumer welfare, especially if agents' qualities, costs, and thus equilibrium profits, are not symmetric. Exploring alternative measures, which could be inspired by social choice theory, is a promising avenue for future research.

4 The collusive strategy landscape

In this section, we discuss how our model's episodic nature and finite inventory significantly affect the strategies for establishing and maintaining learned tacit collusion compared to the previously considered infinite horizon setting. It is common economic intuition (e.g., (Harrington, 2018)) that in order to maintain collusive agreements, agents need to remember past actions and have mechanisms to punish those who deviate from the agreed-upon strategy[2]. Standard punishment strategies include a temporary or permanent shift to a competitive price level after the deviation is detected which results in lower profits for all firms. It has been well documented that learning algorithms converge to these strategies in the infinite horizon setting (Calvano et al., 2020a; Hettich, 2021; Deng, Schiffer, & Bichler, 2024). Such strategies are only credible as long as sufficient time and supply is available for the punishment to offset the short-term gains from a deviation. These conditions are not always met in our settings that lead to new collusive strategies.

Infinite horizon games These settings allow for deriving unique competitive and collusive equilibrium price levels through implicit formulas with the most commonly used Bertrand competition models. They provide the most room for collusive strategies to emerge and sustain since there is no time constraint for a punishment strategy's credibility. Typically, stable collusion manifests in two forms. First, *reward-punishment schemes:* Agents cooperate by default and punish deviations. A deviating agent is punished by others charging competitive prices, thereby removing the benefits of collusion temporarily, until the supra-competitive prices are reinstated. This dynamic involves agents synchronizing over rounds to restore higher price levels after a deviation. This pattern can be observed as fixed, supra-competitive prices and verified by forcing one agent to deviate and recording everyone else's responses. Second, *Edgeworth price cycles:* This pattern involves agents sequentially undercutting each other's prices until one reverts to the collusive price, prompting others to follow, restarting the undercutting cycle (Klein, 2021).

[2] Recent work (Arunachaleswaran et al., 2024) suggests that there can exist stable, collusive equilibria of strategies that do not encode threats. They show that near-monopoly prices can arise if a first-moving agent deploys a no-regret learning algorithm, and the second agent subsequently picks a non-responsive pricing policy.

Episodic games In comparison to the infinite horizon setting, collusive strategies can now emerge in two distinct ways. First, through *intra-episode* action-based communication, where agents gradually raise their prices through signaling within a single episode. Second, through training *across many episodes*, where agents eventually learn policies that implement collusive pricing immediately from the start of each new episode. The latter form is prevalent in oligopolistic settings and possibly explained by learners overfitting their strategies to familiar opponents. When faced with new opponents, collusive agents initially play competitively before reestablishing collusion through continued learning (Eschenbaum, Mellgren, & Zahn, 2022). This robustness result suggests that firms aiming to collude can pre-train their pricing agents separately, needing only (likely legal) alignment on the high-level training setups (e.g., algorithm classes, observation modeling, exploration schedule). In our experiments in Section 5.5, we observe evidence of both types of collusion.

The finite time horizon restricts collusive potential by limiting the efficacy of reward-punishment schemes used in infinite-horizon games to maintain collusion. In a one-shot game ($T = 1$) in our Bertrand setting, there exists a unique Nash equilibrium at the competitive price level, as unilateral deviation from collusive prices is profitable and future punishment is impossible. In the finite horizon case ($T > 1$), the same logic applies at the final period ($t = T$), such that any Nash equilibrium strategy will price competitively in the last timestep. By induction from $t = T$ backwards, this argument extends to all periods $t = T - 1, \ldots, 1$, defining a unique Nash equilibrium where agents compete throughout the episode. Does this mean that collusion in episodic games is impossible? No: If agents remember past interactions across episodes, deviations can be punished in future episodes. Surprisingly, our experiments in Section 5 show that even without cross-episode memory, learning agents in sufficiently long episodes can converge to collusive strategies of the signaling, stable or cyclic kind. We observe that some agents learn to play collusively at episode start and defect toward the end, suggesting that discovering the full backward induction argument through (often random) exploration is unlikely enough in practice.

Episodic, inventory-constrained model Inventory constraints significantly complicate the state and strategy space by making the reward achieved from a pricing strategy dependent on inventory levels. Determining the competitive and collusive price levels becomes more complex because the solution formulas from the Bertrand or Cournot settings require smoothness or convexity assumptions that no longer hold, preventing the standard uniqueness proofs. We approach finding a Nash equilibrium by modeling each episode as a simultaneous-move game where agents set entire price vectors before the episode starts for the complete episode. We provide further details in Section 5.1. We solve the resulting generalized Nash equilibrium problem numerically and prove that its solutions are Nash equilibria in our Markov game. We find that in our model, both the competitive and collusive solutions consist of repeating their prices from the one-period equivalents T times. If agents discount future rewards, both equilibria shift to lower prices and higher profits early in the episode and vice versa toward its end. In addition, price levels remain distinct even with strict inventory constraints. Due to the difficulty in predicting or interpreting observed behavior in this complex setting, we see value in analyzing different types of learners as part of future work.

5 Experiments

In Section 5.1 we first show how to find the competitive and monopolistic price levels needed to calculate the collusion measure defined in Definition 3.2, and how they change under different inventory constraints. Then, we show that PPO (Schulman et al., 2017) and DQN (Mnih et al., 2015), two commonly used deep RL algorithms, can learn to collude in our episodic model. Finally, we analyze their learned strategies and their dependence on hyperparameters.

5.1 Obtaining competitive and collusive equilibrium prices

Previous works' Bertrand settings use analytic formulae to compute Nash equilibrium and monopolistic optimum price vectors p^N and p^M for single-period cases. However, a closed-form solution is not available for our problem setting. We therefore use numerical methods to calculate the competitive and collusive solutions as defined in Definition 3.1 and use these values to define the collusion measure in Definition 3.2.

First, we calculate the profits and prices in the monopolistic (perfectly collusive) setting by assuming a central optimizer who chooses prices for all agents maximizing the total profit. Second, to calculate the same for the competitive Nash equilibrium, we model an entire episode as a *simultaneous-move game (SMG)*, where all agents i must simultaneously decide all T prices in their vector

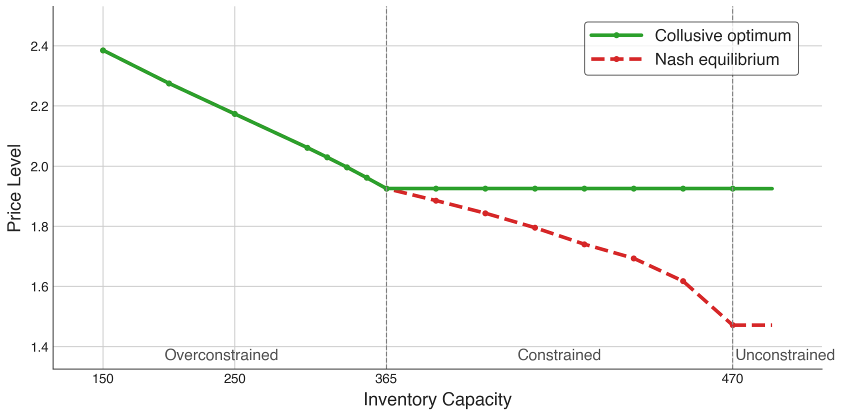


Figure 1: One-period equilibrium price levels as a function of inventory capacity for two equally constrained agents.

$p_i = (p_{i,1}, \dots, p_{i,T})$ before an episode begins. Let $p = (p_1, \dots, p_n)$ encompass all agents' price vectors, with p_{-i} representing all agents' vectors except i's. The solution to this SMG is then a Generalized Nash Equilibrium defined as follows.

Definition 5.1. The *Generalized Nash Equilibrium Problem (GNEP)* consists of finding the price vector $p^* = (p_1^*, \dots, p_n^*)$ such that for each agent i, given p_{-i}^*, the vector p_i^* solves the following inventory-constrained revenue maximization problem

$$\begin{aligned} \max_{p^{(i)}} \quad & \sum_{t=1}^{T} (p_{i,t} - c_i) \lfloor \lambda d_{i,t} \rfloor \\ \text{subject to} \quad & \sum_{t=1}^{T} \lfloor \lambda d_{i,t} \rfloor \leq I, \quad p_i \geq 0. \end{aligned}$$

The solution price vector p^* can be interpreted as the *actions* of a set of agent policies playing an episode of the Markov game. The following lemma shows that a set of policies that result in the price vector p^* form a Nash Equilibrium in the Markov Game.

Lemma 5.2. *Given a Markov Game with deterministic transitions, let $p^* = (p_1^*, \dots, p_n^*)$ be the solution to Definition 5.1 and define $\pi^* = (\pi_1^*, \dots, \pi_n^*)$, as $\pi_i^*(s_t) = p_{i,t}^*$ for all i, t, and $s_t \in \mathcal{S}$. Then π^* is a Nash equilibrium in the Markov Game.*

The full proof can be found in Appendix D. Details of our numerical approach to solving the GNEP are found in Appendix A.

Without discounting, the episodic equilibrium price vectors repeat the single-period equilibrium with the same parameters T times. Figure 1 shows how inventory constraints affect market dynamics. When inventories exceed the demand at the competitive equilibrium, the equilibria correspond to the unconstrained setting. As inventories shrink, the competitive price level rises, as it is harder for firms to undercut and profit from the increased demand. When inventory size matches the demand at the collusive price, the collusive and competitive price levels converge. Further tightening of constraints pushes both coinciding prices higher. In our experiments we choose the constraint's value between the two extremes to allow for differentiation between competitive and collusive behavior and a well-defined collusion index, and investigate the effect of the inventory size on learned collusion in Section 5.6.

5.2 Model parameters

We evaluate the potential for RL algorithms to collude in our model using a duopoly situation with two agents[3]. We use either of two popular algorithms, namely Deep Q-Networks (DQN) (Mnih

[3] Two agents suffice to demonstrate learned collusion in the finite horizon game and the impact of inventory constraints. A duopoly is a reasonable assumption in the ARM domain, as many routes are dominated by 2-3 airlines. For $n > 2$ agents, Abada and Lambin, 2023; Hettich, 2021 show collusion indeed diminishes due to exponential growth in joint policy space hindering joint exploration, but does not fully disappear.

et al., 2015) and Proximal Policy Optimization (PPO) (Schulman et al., 2017) for learning without weight-sharing between agents. Agents represent identical firms, sharing the same qualities $\alpha_i = 2$, marginal costs $c = c_i = 1 \; \forall i$, a horizontal differentiation factor of $\mu = 0.25$, an outside good quality of $\alpha_0 = 0$, and a demand scaling factor of $\lambda = 1000$. For the main results presented in Section 5.4 and Section 5.5, we set the inventory constraints to $440 \cdot T$ and the episode length $T = 20$.

Due to the symmetry between agents, Nash and monopolistic price levels are identical for both of them, and the price levels and the corresponding demands are $p^N = 1.693, p^M = 1.925$ and $d^N = 440 d^M = 365$ for our inventory constrained case. Agents choose prices from a discretized interval $[p^N - \xi(p^M - p^N), p^M + \xi(p^M - p^N)]$ with 15 steps and $\xi = 0.2$, such that the competitive and collusive actions correspond to $a^N = 2$ and $a^M = 12$ respectively. In particular, the price range for our setting is $[1.693, 1.925]$. In Appendix E.3, we provide further results on experiments with a price range defined with the unconstrained Nash equilibrium prices to demonstrate that agents are still capable of learning collusion and their actions quickly converge to the price range defined with the constrained Nash equilibrium prices.

5.3 Training setup

We train our algorithms by playing 1000 and $50,000$ episodes for PPO and DQN, respectively, and updating weights after every episode for PPO or every fourth for DQN. We train 100 pairs of PPO or DQN on unique random seeds (40 for the boxplots). After training, we analyze each agent pair by observing their play in a single episode. This joint training aligns with previous work Calvano et al., 2020a; Koirala and Laine, 2024 and real market situations, where firms learn while competing, updating pricing strategies based on market success. Solid lines and shaded areas in our plots represent the averages and standard deviations of their metrics. For our DQN agent, we use *epsilon-greedy* exploration with an exponentially decaying epsilon, while the PPO agent anneals its entropy coefficient to similarly reduce exploration over time. For evaluation episodes, DQN uses a fully greedy action selection. We normalize the rewards during training to the interval $[0, 1]$ based on minimum and maximum possible values. This makes training slightly more stable. However, collusion is still achieved with unnormalized rewards. A full description of the hyperparameters used for DQN and PPO can be found in Appendix B.2. We use the JAX framework on a custom codebase built on (Willi et al., 2023). Our experiments were run on a compute cluster on a mix of nodes with each run using at most four vCPU cores, 8GB of RAM, and either a NVIDIA T4 or NVIDIA V100 GPU. However, a single run can be done on a consumer laptop (Apple M1 Max, 32GB RAM) in under one hour.

5.4 Analysis of learning process

Figure 2 shows two training runs for DQN and PPO agents. For both algorithms, agents quickly converge to each other and to competition as their learning targets are initially unstable, with high epsilon (DQN) and entropy (PPO) forcing random actions. This makes it hard for agents to adapt to their opponent's underlying policy and leads to them learning the best-response strategy against a random opponent, playing competitively. As the exponentially decaying epsilon and entropy curves flatten and the agents face an increasingly predictable opponent that they can adapt to, they begin colluding. Prices rise gradually and jointly before leveling off at a collusive level. PPO converges in both much fewer episodes and achieves higher levels of collusion, with an average collusion index of $\Delta_e = 0.43$ over the last 10% of episodes, compared to DQN's $\Delta_e = 0.23$. These values, lower than in prior studies in the standard Bertrand setting (Calvano et al., 2020a; Hettich, 2021; Deng, Schiffer, & Bichler, 2024), highlight the greater challenge of collusion in our more complex model. Regulatory efforts could focus on the gradual increase in prices to mitigate algorithmic collusion, which we consider to be an interesting direction for future work.

5.5 Analysis of collusive strategies

After training, we simulate the agents in an evaluation episode (Figure 3). We focus on DQN here, discussing PPO in Appendix E.2. Our DQN agents show behavior that slowly rises in collusiveness until both agents defect near the end of the episode. This suggests that the agents are capable of learning that late defection cannot be punished, while not fully applying the backward induction argument from Section 4. The rise in collusion at the beginning of the episode suggests a capability of establishing *intra-episode* collusion, with the gradual, mutual price increase acting as a form of signaling. In Appendix E.8, we show results without inventory constraints, where the agents' price curve is flatter, suggesting a strategy more based on solidified collusion over multiple episodes.

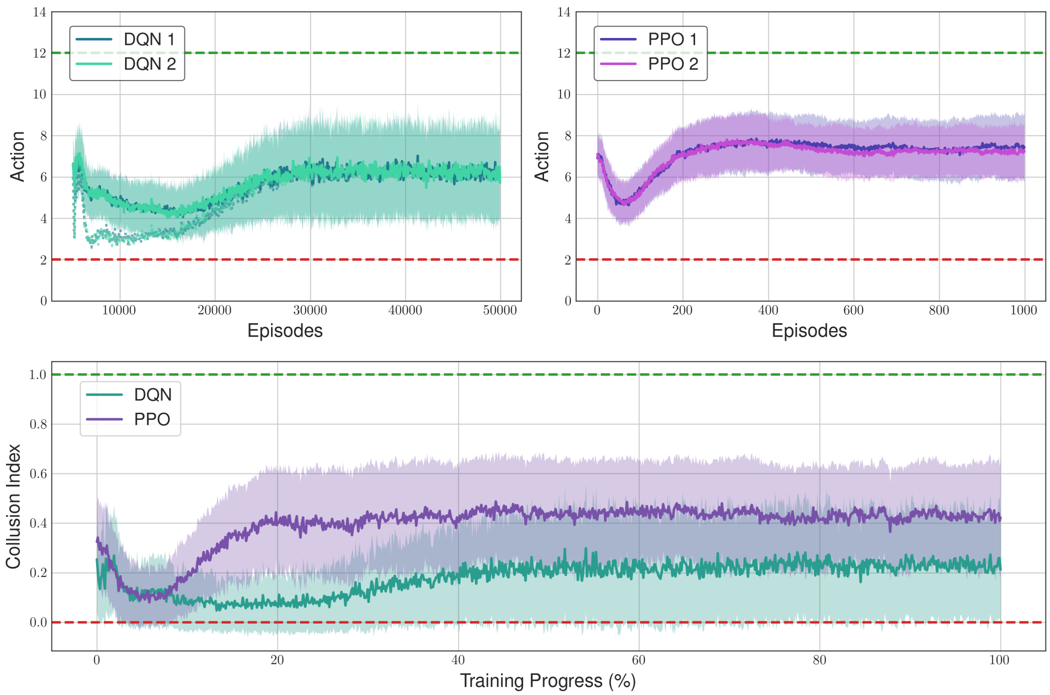


Figure 2: Evolution of training two DQN and two PPO agents in our model, showing average agent actions per episode (DQNs top left, PPOs top right) and collusion index (bottom) with collusive and competitive actions indicated with the green upper and red lower dashed lines respectively. In DQN's training plot, the dotted lines are the greedy actions that DQN would have chosen. Both DQN and PPO first converge to competition before gradually rising toward collusion.

To analyze the nature of the learned strategy, we force one agent to deviate at a certain timestep and record the response by both agents similarly to Calvano et al. (2020a). Interestingly, deviation produces only a small reaction by the competing agent, while the deviating agent quickly returns to near their collusive level. With a deviation at time $t = 1$ or at $t = 9$, the impact on overall episode profits is negligible for both agents, with the deviating agent breaking even and the non-deviating agent losing only 0.2% profit overall. We refer the reader to Appendix E.9 for details.

Figure 4 shows the best-response surfaces of the first agent at different points in the episode, with a remaining inventory linearly interpolated from full to none over the episode (corresponding to the agents' evaluated strategy) and averaged over 100 trained agent pairs. We make two observations.

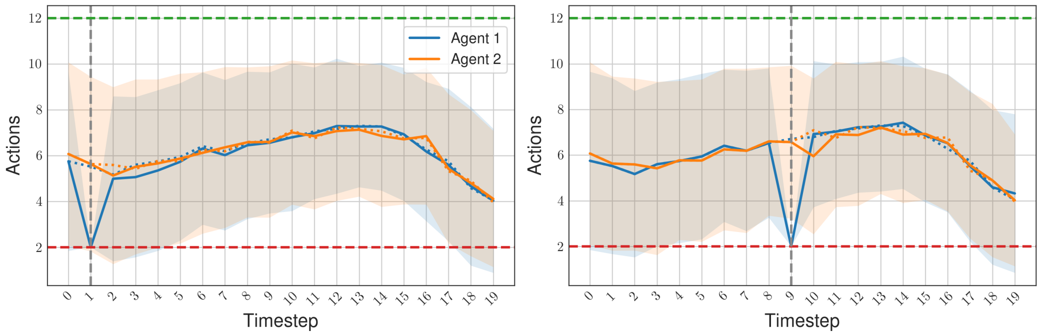


Figure 3: Behavior of two DQN agents during an episode after forcing one agent to deviate at time $t = 1$ and $t = 9$ respectively. Dotted lines indicate evolution without deviation. Deviations provoke a competitive reaction, with both agents quickly returning to collusion.

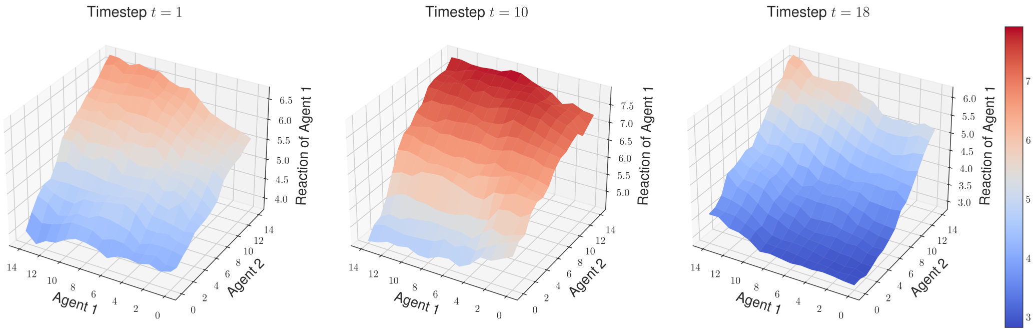


Figure 4: The surfaces show a DQN agent 1's learned best response under their greedy policy (i.e., the action with the highest Q-value) to a state given by both agent's prices (x- and y-axes), timestep and symmetric remaining inventory level.

First, the agent always punishes opponent deviations by pricing lower than the previous price level. Second, at the beginning and end of the episode, the agent's best-response surface shows some symmetry indicative of more competitive behavior. There, the agent will react to their own deviations by pricing even lower in consecutive periods, anticipating a 'price war'. During the middle of the episode, the agent instead returns to previous or even higher collusion levels after own defections, signaling cooperation, and punishes opponent deviations with slight undercutting. Near the end of the episode, they shift to more competitive behavior, punishing deviations more strongly. This topology suggests that if both agents start near the competitive equilibrium, they will both react in a way that jointly 'climbs the hill' to collusion, leveling out at an action of roughly 7 as indicated by the flat top. The second agent behaves similarly. These results suggest that DQN agents are well aware of competitive strategies and choose to collude in a robust way reliant on rewards and punishments. Appendices E.4 and E.5 contain results for uneven inventory constraints and limiting observability of opponent inventory and time, neither of which significantly hinder the emergence of collusion.

5.6 Hyper- and environment parameters

We analyze the impact of changing agent hyperparameters and environment characteristics on the convergence and collusive tendencies of DQN and PPO agents. We show comparisons for agent learning rate, inventory constraint, and episode length here, with additional results deferred to Appendix E.10. To judge the convergence of two agents toward each other throughout the training run, we use the following metric:

$$\frac{1}{0.1E} \sum_{e=0.9E}^{E} \frac{1}{T} \sum_{t=1}^{T} \frac{|p_{0,t} - p_{1,t}|}{p^M - p^N}$$

adapted from Deng, Schiffer, and Bichler (2024), where E is the number of training episodes. It takes the average difference of both agents' prices across an episode relative to the width of the Nash-monopolistic price interval. Values below 0.2 are interpreted as converged.

In our analysis, we vary single parameters from the reference setup described in Section 5.2, train agents on 40 different seeds, and for each parameter value, record the distribution of convergence metric and collusion index over those seeds, averaged over the last 10% of training run episodes.

Learning rate is perhaps the most important agent parameter, as it regulates the impact of all other agent parameters. Section 5.6 demonstrates that both PPO and DQN agents achieve better convergence and increased tendency to compete at lower learning rates. The reduced ability to adapt to an opponent's strategy still allows agents to learn the opponent-independent best-response of competition at initial training episodes, but attempts to establish the gradual, mutual increase in price seen in Figure 2 happen more rarely and revert to competition more often. A higher learning rate does not translate to more likely collusion, as the increased ability to adapt to an opponent is balanced by the potential to overreact to the opponent's random actions. Overall, collusion and convergence appear to be robust to moderate changes in learning rate.

We compare metrics among different initial inventory sizes in Figure 6a. Inventory sizes shown are per-timestep; a value of 440 represents a total inventory size of $440 \cdot T$, which we use for the

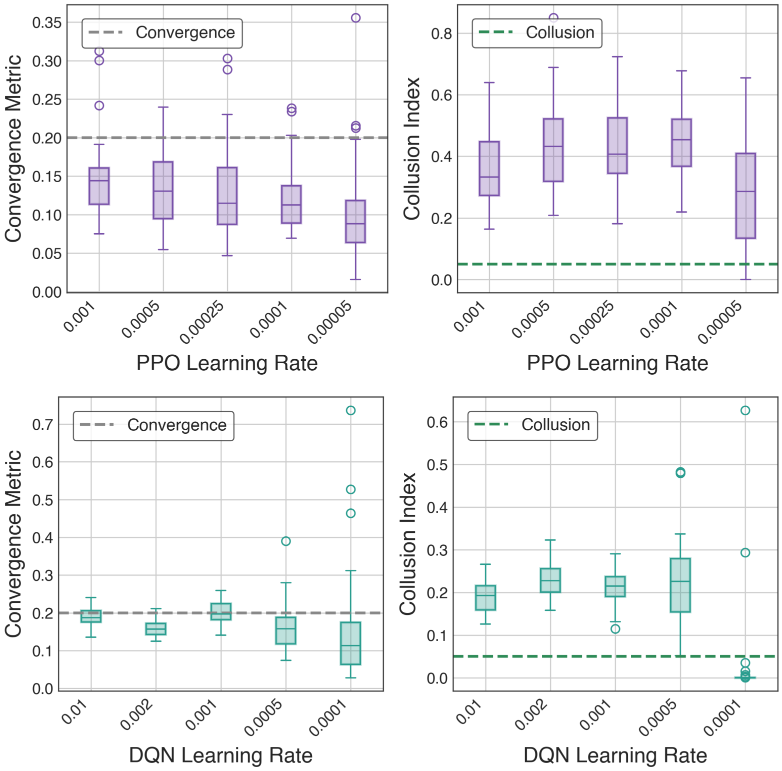
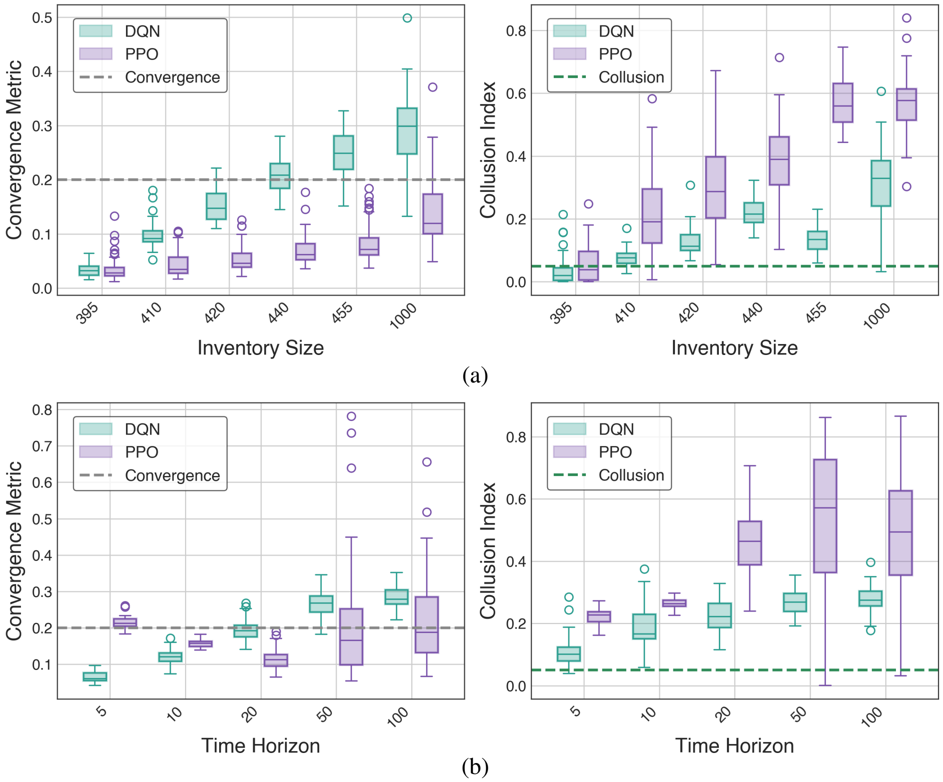


Figure 5: Convergence and collusion metrics for DQN and PPO training runs with varied learning rate. Collusion is robust against varying (yet sufficiently large) learning rate.

other results. Smaller inventories show better convergence and more competitive behavior for both PPO and DQN. This has geometric intuition (cf. Appendix E.6): visualize each agent's reward landscape as a surface over the grid of both agents' prices. Each agent tries to climb toward their peak on the side of the grid's diagonal where they undercut their opponent. Steps toward their peak along their axis harm their opponent. To achieve collusion, agents must jointly climb the ridge along the diagonal of the grid where their landscapes intersect. The closer the two agent's peaks are to the monopolistic optimum on the diagonal, the smaller their incentive to deviate and the smaller the negative impact on their opponent from deviation, easing cooperation. Decreasing inventory capacities reduces the range of prices that agents are incentivized to use as the Nash equilibrium price approaches the monopolistic price. In this "zoomed in" part of the price grid, the peaks now appear further away from each other, making the coordination problem harder.

Figure 6b shows the effect of changing episode lengths. As conjectured in Section 4, longer episodes increase collusion tendencies for both types of learners by providing more opportunities to punish deviations. While PPO's convergence is unaffected, DQN's convergence suffers. This is expected, as DQN generally scales worse to larger state spaces than PPO. It relies on accurately estimating the expected reward for each state-action pair and sufficiently exploring the state space, which becomes harder as that space grows.

We identified additional hyperparameters affecting collusion, such as PPO's number of training epochs (higher increases collusion) and DQN's buffer size (larger increases collusion), shown in Appendix E.10. It is possible to hinder collusion by introducing instability in learning targets, e.g., by filling DQN's buffer or PPO's rollouts with experiences gathered from 'parallel environments'. This parallelization is commonly done to increase training speed on accelerator hardware, but has a concrete impact in this model. We demonstrate this with PPO in Appendix E.7.



(a)

(b)

Figure 6: Convergence and collusion metrics for DQN and PPO training runs with varied inventory sizes (a), and episode time horizons (b). Initial inventory size is the value shown, times the time horizon T. Longer episodes show less reliable convergence, higher potential collusion due to more effective punishment strategies.

6 Conclusion

We formulate price competition between producers as an episodic Markov game motivated by Airline Revenue Management (ARM) and facilitating the analysis of tacit collusion within a finite time horizon and inventory-constrained markets. We propose numerical methods to find competitive and collusive solutions in our model due to the lack of analytical solutions and define a collusion metric based on the total profit achieved in a full episode. Our analysis shows that collusion consistently emerges between independent DQN and PPO algorithms after a brief period of competition and that trained agents quickly revert back to collusive prices after a forced deviation. The proven collusive potential of RL agents in our setting covering many real markets reinforces the call for the development of mitigation strategies and regulatory efforts (Calvano et al., 2020b).

We see our work as a first step toward understanding pricing competition in markets like airline tickets, hotels, and perishable goods with future research directions in extending our Markov Game model to domain specifics. Additionally, we see a need to consider multi-agent specific algorithms, e.g., opponent-shaping agents (Souly et al., 2023), that could establish stronger collusion or even exploit market participants, significantly harming social welfare.

References

Abada, Ibrahim and Xavier Lambin (2023). “Artificial Intelligence: Can Seemingly Collusive Outcomes Be Avoided?” In: *Management Science* 69.9, pp. 5042–5065.

Abada, Ibrahim et al. (2024). “Algorithmic Collusion: Where Are We and Where Should We Be Going?”

Acuna-Agost, Rodrigo, Eoin Thomas, and Alix Lhéritier (2023). “Price Elasticity Estimation for Deep Learning-Based Choice Models:An Application to Air Itinerary Choices.” In: *Artificial Intelligence and Machine Learning in the Travel Industry: Simplifying Complex Decision Making*. Ed. by Ben Vinod. Springer Nature Switzerland.

Alamdari, Neda Etebari and Gilles Savard (2021). “Deep Reinforcement Learning in Seat Inventory Control Problem: An Action Generation Approach”. In: *Journal of Revenue and Pricing Management* 20.5, pp. 566–579.

Arunachaleswaran, Eshwar Ram et al. (2024). “Algorithmic Collusion Without Threats”. arXiv: `2409.03956`.

Asker, John, Chaim Fershtman, and Ariel Pakes (2022). “Artificial Intelligence, Algorithm Design, and Pricing”. In: *AEA Papers and Proceedings* 112, pp. 452–56.

Assad, Stephanie et al. (2024). “Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market”. In: *Journal of Political Economy* 132.3, pp. 723–771.

Ausubel, Lawrence M (1991). “The failure of competition in the credit card market”. In: *The American Economic Review*, pp. 50–81.

Belobaba, Peter Paul (1987). “Air Travel Demand and Airline Seat Inventory Management”. In: *Flight Transportation Laboratory Reports*.

Beneke, Francisco and Mark-Oliver Mackenrodt (2021). “Remedies for Algorithmic Tacit Collusion”. In: *Journal of Antitrust Enforcement* 9.1, pp. 152–176.

Bertrand, Joseph Louis François (1883). “Review of “Theorie mathematique de la richesse sociale” and of “Recherches sur les principles mathematiques de la theorie des richesses.””

Bertsimas, Dimitris and Sanne de Boer (2005). “Simulation-Based Booking Limits for Airline Revenue Management”. In: *Operations Research* 53.1, pp. 90–106.

Bondoux, Nicolas et al. (2020). “Reinforcement Learning Applied to Airline Revenue Management”. In: *Journal of Revenue and Pricing Management* 19.5, pp. 332–348.

Borenstein, Severin and Nancy L Rose (1994). “Competition and price dispersion in the US airline industry”. In: *Journal of Political Economy* 102.4, pp. 653–683.

Brero, Gianluca et al. (2022). “Learning to Mitigate AI Collusion on Economic Platforms”. In: *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 35, pp. 37892–37904.

Bront, Juan José Miranda, Isabel Méndez-Díaz, and Gustavo Vulcano (2009). “A Column Generation Algorithm for Choice-Based Network Revenue Management”. In: *Operations Research* 57.3, pp. 769–784.

Bundeskartellamt and Autorité de la Concurrence (2019). *Algorithms and Competition*. Tech. rep.

Buşoniu, Lucian, Robert Babuška, and Bart De Schutter (2010). “Multi-agent reinforcement learning: An overview”. In: *Innovations in multi-agent systems and applications-1*, pp. 183–221.

Calvano, Emilio et al. (2020a). “Artificial Intelligence, Algorithmic Pricing, and Collusion”. In: *American Economic Review* 110.10, pp. 3267–3297.

Calvano, Emilio et al. (2020b). “Protecting Consumers from Collusive Prices Due to AI”. In: *Science* 370.6520.

Deng, Shidi, Maximilian Schiffer, and Martin Bichler (2024). “Algorithmic Collusion in Dynamic Pricing with Deep Reinforcement Learning”. arXiv: `2406.02437`.

Dinneweth, Joris et al. (2022). “Multi-Agent Reinforcement Learning for Autonomous Vehicles: A Survey”. In: *Autonomous Intelligent Systems* 2.1, p. 27.

Directorate-General for Competition (European Commission) et al. (2019). *Competition Policy for the Digital Era*. Publications Office of the European Union. ISBN: 978-92-76-01946-6.

Eschenbaum, Nicolas, Filip Mellgren, and Philipp Zahn (2022). “Robust Algorithmic Collusion”. arXiv: `2201.00345`.

European Union (2012). “Treaty on the Functioning of the European Union”. Arts. 101-109. `http://data.europa.eu/eli/treaty/tfeu_2012/oj`.

– (2019). “Regulation (EU) 2019/712 on safeguarding competition in air transport, and repealing Regulation (EC) No 868/2004”. `http://data.europa.eu/eli/reg/2019/712/oj`.

Facchinei, Francisco and Christian Kanzow (2007). “Generalized Nash Equilibrium Problems”. In: *4OR* 5.3, pp. 173–210.

Genesove, David and Wallace P Mullin (2001). “Rules, communication, and collusion: Narrative evidence from the sugar institute case”. In: *American Economic Review* 91.3, pp. 379–398.

Gosavi, Abhijit, Naveen Bandla, and Tapas K. Das (2002). “A Reinforcement Learning Approach to a Single Leg Airline Revenue Management Problem with Multiple Fare Classes and Overbooking”. In: *IIE Transactions* 34.9, pp. 729–742.

Gronauer, Sven and Klaus Diepold (2022). “Multi-agent deep reinforcement learning: a survey”. In: *Artificial Intelligence Review* 55.2, pp. 895–943.

Harrington, Joseph E. (2018). “Developing Competition Law for Collusion by Autonomous Artificial Agents”. In: *Journal of Competition Law & Economics* 14.3, pp. 331–363.

Hettich, Matthias (2021). “Algorithmic Collusion: Insights from Deep Learning”. In: *SSRN Electronic Journal*.

Kastius, Alexander and Rainer Schlosser (2022). “Dynamic Pricing under Competition Using Reinforcement Learning”. In: *Journal of Revenue and Pricing Management* 21.1, pp. 50–63.

Klein, Timo (2021). “Autonomous algorithmic collusion: Q-learning under sequential pricing”. In: *The RAND Journal of Economics* 52.3, pp. 538–558.

Koenigsberg, Oded, Eitan Muller, and Naufel Vilcassim (2004). *EasyJet Airlines: Small, Lean and with Prices that Increase over Time*. Working Paper. London Business School Centre for Marketing. URL: `https://lbsresearch.london.edu/id/eprint/3369/`.

Koirala, Pravesh and Forrest Laine (2024). “Algorithmic Collusion in a Two-Sided Market: A Rideshare Example”. arXiv: `2405.02835`.

Lawhead, Ryan J. and Abhijit Gosavi (2019). “A Bounded Actor–Critic Reinforcement Learning Algorithm Applied to Airline Revenue Management”. In: *Engineering Applications of Artificial Intelligence* 82, pp. 252–262.

Littman, Michael L. (1994). “Markov Games as a Framework for Multi-Agent Reinforcement Learning”. In: *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*. ICML’94, pp. 157–163.

Mnih, Volodymyr et al. (2015). “Human-Level Control through Deep Reinforcement Learning”. In: *Nature* 518.7540, pp. 529–533.

Musolff, Leon (2022). “Algorithmic pricing facilitates tacit collusion: Evidence from e-commerce”. In: *Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 32–33.

Ohlhausen, Maureen K. (2017). *Should We Fear The Things That Go Beep In the Night? Some Initial Thoughts on the Intersection of Antitrust Law and Algorithmic Pricing*. Tech. rep. Federal Trade Commission.

Rana, Rupal and Fernando S. Oliveira (2014). “Real-Time Dynamic Pricing in a Non-Stationary Environment Using Model-Free Reinforcement Learning”. In: *Omega* 47, pp. 116–126.

– (2015). “Dynamic Pricing Policies for Interdependent Perishable Products or Services Using Reinforcement Learning”. In: *Expert Systems with Applications* 42.1, pp. 426–436.

Razzaghi, Pouria et al. (2022). “A Survey on Reinforcement Learning in Aviation Applications”. arXiv: `2211.02147`.

Sanchez-Cartas, J. Manuel and Evangelos Katsamakas (2022). “Artificial Intelligence, Algorithmic Competition and Market Structures”. In: *IEEE Access* 10, pp. 10575–10584.

Schulman, John et al. (2017). “Proximal Policy Optimization Algorithms”. arXiv: `1707.06347`.

Shihab, Syed A.M. and Peng Wei (2022). “A Deep Reinforcement Learning Approach to Seat Inventory Control for Airline Revenue Management”. In: *Journal of Revenue and Pricing Management* 21.2, pp. 183–199.

Silver, David et al. (2017). “Mastering the Game of Go without Human Knowledge”. In: *Nature* 550.7676, pp. 354–359.

Silver, David et al. (2018). “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362.6419, pp. 1140–1144.

Souly, Alexandra et al. (2023). “Leading the Pack: N-player Opponent Shaping”. arXiv: `2312.12564`.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement Learning: An Introduction*. MIT press.

Talluri, Kalyan T. and Garrett J. Van Ryzin (2004). *The Theory and Practice of Revenue Management*. Vol. 68. International Series in Operations Research & Management Science. Springer US.

van Ryzin, Garrett and Jeff McGill (2000). “Revenue Management Without Forecasting or Optimization: An Adaptive Algorithm for Determining Airline Seat Protection Levels”. In: *Management Science* 46.6, pp. 760–775.

Waltman, Ludo and Uzay Kaymak (2008). “Q-Learning Agents in a Cournot Oligopoly Model”. In: *Journal of Economic Dynamics and Control* 32.10, pp. 3275–3293.

Wang, Rui et al. (2021). “Solving a Joint Pricing and Inventory Control Problem for Perishables via Deep Reinforcement Learning”. In: *Complexity*.

Willi, Timon et al. (2023). “Pax: Scalable Opponent Shaping in JAX”. `https://github.com/ucl-dark/pax`.

Wong, Annie et al. (2023). “Deep multiagent reinforcement learning: Challenges and directions”. In: *Artificial Intelligence Review* 56.6, pp. 5023–5056.

Yang, Yaodong and Jun Wang (2021). “An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective”. arXiv: `2011.00583`.

A Numerical solution strategy for Nash & monopolistic prices

To solve the GNEP for competitive equilibrium prices, we use a Gauss-Seidel-type iterative method (Facchinei & Kanzow, 2007). We start with an initial price vector guess and proceed through a loop where each iteration updates each agent's price by solving their subproblem. For agent i at iteration k, it uses the fixed opponent prices from the latest estimate. The process repeats until convergence to p^*. Each agent's subproblem is a mixed-integer, nonlinear optimization problem (MINLP), with neither convex objectives nor constraints. We use *Bonmin*, a local solver capable of handling larger instances at the risk of missing global optima. We mitigate this by initiating the solver from multiple different starting points. For the collusive optimum, we simulate a scenario where one agent sells n items, aiming to maximize the total episodic revenue under n inventory constraints. This problem is again a non-convex MINLP. Our implementation uses the open-source COIN-OR solvers via Pyomo in Python.

B Parameters used for training, environment, DQN and PPO

B.1 Environment and agent parameters

All our experiments use identical parameter values for both agents.

Parameter	**Value**
Product quality α_i	2
Outside good quality α_0	0
Marginal cost c_i	1
Horizontal differentiation μ	0.25
Time horizon T	20
Demand scaling factor λ	1000
Inventory capacity I_i	$440 * T$
Nash price p^N (unconstrained)	1.471
Nash price p^N (constrained)	1.693
Monopolistic price p^M	1.925
Number of prices in interval	15
Price interval parameter ξ	0.2

Table 1: Environment and agent parameters

B.2 DQN and PPO hyperparameters

We used the same neural network architecture for both DQN and PPO, of 2 hidden layers with 64 neurons each. These hyperparameters were found by starting with generally accepted values from reference implementations and refined by doing grid searches over up to three parameters at a time.

DQN's epsilon-greedy strategy's epsilon parameter is annealed via an exponential decay from an initial value of $\epsilon_{\max} = 1$ to $\epsilon_{\text{min}} = 0.015$ at the end of the training run.
At training episode $e \in \{0, \ldots, E\}$, epsilon's value is $\epsilon_{\max} * \left(\frac{\epsilon_{\text{max}}}{\epsilon_{\text{min}}}\right)^{e/E}$.

Parameter	**Value**
Training episodes E	$50\,000$
Learning rate	0.001
Adam epsilon	0.001
Epsilon-greedy (annealed) ϵ_{min}	0.015
Replay buffer size	$200\,000$
Replay buffer batch size	64
Gradient norm clipping	25
Initial episodes without training	5000
Train agent every ... episodes	4
Target network update every ... episodes	200
Network layer sizes	$[64, 64]$

Table 2: DQN hyperparameters

PPO's entropy coefficient parameter is annealed via an exponential decay from an initial value of $\text{ent}_{\text{max}} = 0.03$ to $\text{ent}_{\text{min}} = 0.0001$ at 75% of the training run (and is clipped to ent_{min} afterwards). At training episode $e \in \{0, \ldots, E\}$, the coefficient's value is $\text{ent}_{\text{max}} * \left(\frac{\text{ent}_{\text{min}}}{\text{ent}_{\text{max}}}\right)^{e/0.75E}$.

Parameter	**Value**
Training episodes E	1000
Learning rate	2.5×10^{-4}
Adam epsilon	1×10^{-5}
Number of minibatches	10
Number of training epochs	20
GAE-lambda	0.95
Value coefficient (with clipping)	0.5
Gradient norm clipping	0.5
Network layer sizes	$[64, 64]$

Table 3: PPO hyperparameters

C Literature review

Examples and description of tacit collusion Firms across various sectors, from insurance to flight tickets, employ *algorithmic pricing* to maximize revenue by leveraging data on market conditions, customer profiles, and other factors. These algorithms' growing complexity raises challenges for maintaining fair competition and detect firms that *tacitly collude*, ones which jointly set *supra-competitive* prices (i.e., above the competitive level) or limit production *without explicit agreements or communication*. Recently, evidence has emerged that companies are already using algorithmic pricing to inflate prices market-wide at the cost of consumers. For instance, Assad et al. (2024) showed that German fuel retailer margins increased by 38% following the widespread adoption of algorithmic pricing. Other examples are found in setting credit card interest rates (Ausubel, 1991) and consumer goods markets (Genesove & Mullin, 2001).

Legal developments around algorithmic collusion Current anti-collusion policies mainly address explicit agreements, making tacit collusion inferred from company behaviors rather than evidence of an agreement, more elusive to prove. There is growing concern among regulators (Ohlhausen, 2017; Bundeskartellamt & Autorité de la Concurrence, 2019; Directorate-General for Competition (European Commission) et al., 2019) and researchers (Harrington, 2018; Beneke & Mackenrodt, 2021; Brero et al., 2022) that AI-based pricing algorithms might evade competition laws by colluding tacitly, without direct communication or explicit instruction during learning. This highlights the need for better strategies to prevent collusion or mitigate its negative effects on the market.

Reinforcement learning (RL) background *Reinforcement learning* (Sutton & Barto, 2018) is an advanced segment of machine learning where agents learn to make sequential decisions by interacting with an environment. Unlike traditional machine learning methods which rely on static datasets, RL emphasizes the development of autonomous agents that improve their behavior through trial-and-error, learning from their own experiences. This approach enables agents to understand complex patterns and make optimized decisions in scenarios with uncertain or shifting underlying dynamics. *Multi-agent* RL extends this concept to scenarios involving multiple decision-makers, each optimizing their strategies while interacting with others and the environment (Buşoniu, Babuška, & De Schutter, 2010). In MARL settings, agents can be incentivized to behave competitively, as seen in zero-sum games like Go (Silver et al., 2017, 2018), cooperatively, like in autonomous vehicle coordination (Dinneweth et al., 2022) or a mix of the two that includes our problem, i.e., markets and pricing games. MARL, while posing challenges such as *non-stationarity* and *scalability*, enables agents to adapt to and influence competitors' strategies, facilitating tacit collusion.

Collusion & regulation in airline revenue management (ARM) Originally a strictly regulated sector with price controls, ARM was deregulated in 1978 in the US and Europe, leading to a competitive landscape of private carriers whose pricing strategies are subject only to general laws against anti-competitive behavior (European Union, 2012)(Art. 101-109). However, this deregulation has caused market consolidation, prompting regulatory responses to protect competition (European Union, 2019). Even prior to algorithmic pricing, regulators have identified pricing behaviors suggestive of tacit collusion (Borenstein & Rose, 1994), underscoring the challenge of distinguishing between collusive behavior and independent but parallel responses to market conditions.

Background on the field of revenue management (RM) Each of the agents that we model is individually maximizing their revenue, relating our work to the field of *revenue management (RM)* (Talluri & Van Ryzin, 2004). As a competitive market with slim net margins, airlines are increasingly turning to *dynamic pricing* (Koenigsberg, Muller, & Vilcassim, 2004) beyond traditional *quantity-based* and *price-based* RM, replacing the hugely popular expected marginal seat revenue (EMSR) models (Belobaba, 1987). Our problem falls into the price-based RM category, even though we do model aspects of capacity management with our inventory constraints. In quantity-based RM, agents decide on a production quantity with the price for their good being the result of a market-wide fixed function of that decision, and models often impose no limit on the offered quantity. In our model, agents decide their price, and demand results from a market-wide function. Our aim is that agents learn to predict the impact of their pricing choices on the demand and thus sold quantity, in order to optimally use their constrained inventory.

Learning in general RM In recent years, reinforcement learning agents have seen increased use in revenue management outside of the airline context. Examples include learning both pricing and production quantity strategies in a market with perishable goods (Wang et al., 2021), producing a pricing policy by learning demand (Rana & Oliveira, 2014, 2015) and analyzing the performance of different popular single-agent RL in various market settings (Kastius & Schlosser, 2022) (here Q-learning and Actor-Critic). The use of largely uninterpretable learned choice or pricing models introduces new challenges, such as deriving economic figures like the elasticity of demand with respect to price (Acuna-Agost, Thomas, & Lhéritier, 2023).

Learning in ARM While early work used e.g. heuristically solved linear programming formulations (Bront, Méndez-Díaz, & Vulcano, 2009) or custom learning procedures (van Ryzin & McGill, 2000; Bertsimas & de Boer, 2005), recent studies have explored single-agent reinforcement learning in ARM to learn optimal pricing (Razzaghi et al., 2022). These model the problem as a single-agent Markov decision problem (MDP) (Gosavi, Bandla, & Das, 2002; Lawhead & Gosavi, 2019) and consider realistic features like cancellations and overbooking (Shihab & Wei, 2022). The application of *deep reinforcement learning (deep-RL)* (Mnih et al., 2015) is growing in this complex market (Bondoux et al., 2020; Alamdari & Savard, 2021), but these models often overlook the multi-agent nature of the airline market. We model the market as a multi-agent system with individual multi-agent learners, a critical yet unexplored aspect in current research (Razzaghi et al., 2022).

D Proof of Lemma 1

Proof. Let us introduce some terminology first.

Definition D.1. Fix an agent i with policy π_i or price vector $p^{(i)}$, and fix opponent policies $\pi^{(-i)}$ or prices $p^{(-i)}$.

- A *useful deviation* is a policy π_i' or price vector $p^{(i)'}$ that strictly increases i's revenue over the whole episode compared to playing π_i or $p^{(i)}$. We use this term in both the Markov game and SMG.
- We call a price vector $p^{(i)} = (p_{i,1}, \ldots, p_{i,T})$ *feasible in the GNEP* if it fulfills the inventory constraint of i's revenue maximization problem in Definition 5.1, and *infeasible in the GNEP* if it does not.
- We call a policy π_i *simple*, if at each time t, it outputs the same value for all states s_t, i.e. $\forall t \, \forall s_t : \pi_i(s_t) \equiv \text{const}_t$.

Intuitively, we construct a set of simple policies where each agent always plays their GNEP solution, no matter the state, and show that this set of policies is a Nash equilibrium.

First, observe that those simple policies result in the same set of price vectors p^* in every evolution of the Markov game. In particular, fixing opponent strategies $\pi^{(-i)*}$ results in agent i facing the same fixed opponent price vectors $p^{(-i)*}$ (from the GNEP solution) in every evolution of the Markov game. Therefore, to prove that π^* is a Nash equilibrium in the Markov game it is enough to prove that for any agent i and fixed opponent price vectors $p^{(-i)*}$, there does not exist a useful deviation price vector $p^{(i)'} \neq p^{(i)}$. If a useful deviation policy π_i' existed for i, in at least one timestep t it would have to pick a price $p_{i,t}' \neq p_{i,t}$, so by ruling out a useful price vector deviation we also rule out a useful policy deviation.

Claim: Let $p^{(-i)}$ be fixed opponent price vectors. Given any price vector $p^{(i)}$ for agent i, there always exists a price vector $\bar{p}^{(i)}$ that is feasible in the GNEP and such that playing $\bar{p}^{(i)}$ results in revenue for i that is as great as or greater than that from playing $p^{(i)}$.

Given opponent prices $p^{(-i)*}$, if a useful deviation $p^{(i)'} \neq p^{(i)*}$ exists for agent i, it must be infeasible in the GNEP (otherwise $p^{(i)*}$ wouldn't be a revenue-maximizing solution to agent i's GNEP's subproblem). However, since the claim implies that we could construct a $\bar{p}^{(i)}$ that is feasible in the GNEP and has equivalent revenue for i as the infeasible $p^{(i)'}$, it would be a useful deviation for agent i in the SMG to play $\bar{p}^{(i)}$ given $p^{(-i)*}$, contradicting the assumption that p^* is a NE.

Proof of Claim: Let opponent prices be fixed $p^{(-i)}$. Let $p^{(i)}$ a price vector in the Markov game that's infeasible in the GNEP (otherwise we're trivially done). Let i's inventory at t be x_t. Let $\hat{t} \in \{1, \dots, T\}$ be the *sell-out time*, i.e., the last timestep in which i has nonzero inventory, meaning $\hat{t} := \max\{t \in \{1, \dots, T\} | x_{\hat{t}} > 0\}$ such that $x_{\hat{t}} = 0$ and $\forall t > \hat{t} : x_t = 0$. Let $d(p_{i,t}, p_{(-i),t}) := \lfloor \lambda d_{i,t} \rfloor$ be the scaled, truncated MNL demand of agent i at time t given price vector p, which is a decreasing function in $p_{i,t}$.

Define

$$\begin{aligned}
\bar{p}_{i,\hat{t}} &:= \sup\{q \mid d(q, p_{(-i),\hat{t}}) = x_{\hat{t}}\} \\
\bar{p}_{i,t} &\in \{q \mid d(q, p_{(-i),t}) = 0\} \quad \forall t > \hat{t}.
\end{aligned}$$

Then, let $\bar{p}^{(i)} := (p_{i,1}, \dots, p_{i,\hat{t}-1}, \bar{p}_{i,\hat{t}}, \bar{p}_{i,\hat{t}+1}, \dots, \bar{p}_{i,T})$.

Given the other agents' fixed price vectors $p^{(-i)}$, the vector $\bar{p}^{(i)}$ is feasible in the GNEP. To see this, consider that every price vector has a sell-out time $\hat{t}$. At any point in time before $\hat{t}$, the accumulated demand up until that time is lower than inventory, otherwise $\hat{t}$ wouldn't actually be the sell-out time. The GNEP's feasibility constraint is only violated if at $\hat{t}$, demand is larger than remaining inventory $x_{\hat{t}}$, or if at any $t > \hat{t}$, demand is larger than 0. The construction of $\bar{p}^{(i)}$ ensures that it has the same sell-out time $\hat{t}$, and the construction of $\bar{p}_{i,t}$ for $t \geq \hat{t}$ ensures that demand at $\hat{t}$ matches inventory left, and that demand at $t > \hat{t}$ is zero, meaning that $\bar{p}^{(i)}$ cannot violate the feasibility constraint.

Now we just need to prove that given fixed opponent prices $p^{(-i)}$, agent i's reward in the Markov game when playing $\bar{p}^{(i)}$ is as great as or greater than their reward when playing $p^{(i)}$. Their reward when playing $p^{(i)}$ is given by

$$\begin{aligned}
&\Sigma_{t=1}^{\hat{t}-1}(p_{i,t} - c) \min\left(d(p_{i,t}, p_{(-i),t}), x_t\right) \\
&+ (p_{i,\hat{t}} - c) \min\left(d(p_{i,\hat{t}}, p_{(-i),\hat{t}}), x_{\hat{t}}\right) \\
&+ \Sigma_{t=\hat{t}+1}^{T}(p_{i,t} - c) \min\left(d(p_{i,t}, p_{(-i),t}), x_t\right)
\end{aligned}$$

We now replace $p^{(i)}$ with $\bar{p}^{(i)}$ and compare each term.

In the *first term*, as we know that for $t < \hat{t}$ i's demand is always lower than their inventory by definition of $\hat{t}$, the term reduces to

$$\Sigma_{t=1}^{\hat{t}-1}(p_{i,t} - c) d(p_{i,t}, p_{(-i),t}).$$

Since $p_t = \bar{p}_t$, we see that the first revenue term's value stays equal:

$$\begin{aligned}
&\Sigma_{t=1}^{\hat{t}-1}(p_{i,t} - c) \min\left(d(p_{i,t}, p_{(-i),t}), x_t\right) \\
&= \Sigma_{t=1}^{\hat{t}-1}(p_{i,t} - c) d(p_{i,t}, p_{(-i),t}) \\
&= \Sigma_{t=1}^{\hat{t}-1}(\bar{p}_{i,t} - c) d(\bar{p}_{i,t}, p_{(-i),t}).
\end{aligned}$$

In the *second term*, by definition of $\hat{t}$, we know that

$$\min\Big(d(p_{i,\hat{t}}, p_{(-i),\hat{t}}), x_{\hat{t}}\Big) = d(p_{i,\hat{t}}, p_{(-i),\hat{t}}) = x_{\hat{t}},$$

thus the term reduces to

$$(p_{i,\hat{t}} - c)d(p_{i,\hat{t}}, p_{(-i),\hat{t}}).$$

Since $d(p_{i,\hat{t}}, p_{(-i),\hat{t}}) \geq x_{\hat{t}}$, and by construction $d(\bar{p}_{i,\hat{t}}, p_{(-i),\hat{t}}) = x_{\hat{t}}$, and $d(\cdot, p_{(-i),\hat{t}})$ decreasing, we get $\bar{p}_{i,\hat{t}} \geq p_{i,\hat{t}}$. We also know that i will always choose a price $\geq c$ to ensure non-negative revenue. Thus, we see that the second revenue term's value can only increase:

$$\begin{aligned}
&(p_{i,\hat{t}} - c)\min\Big(d(p_{i,\hat{t}}, p_{(-i),\hat{t}}), x_{\hat{t}}\Big)\\
&= (p_{i,\hat{t}} - c)d(p_{i,\hat{t}}, p_{(-i),\hat{t}})\\
&\leq (\bar{p}_{i,\hat{t}} - c)d(\bar{p}_{i,\hat{t}}, p_{(-i),\hat{t}}).
\end{aligned}$$

In the *third term*, by definition of $\hat{t}$, we know that $\forall t > \hat{t} : x_t = 0$, and since by construction of $\bar{p}^{(i)}$ we also know that $\forall t > \hat{t} : d(\bar{p}_{i,t}, p_{(-i),t}) = 0$, we see that the term's value remains zero:

$$\begin{aligned}
&\Sigma_{t=\hat{t}+1}^{T}(p_{i,t} - c)\min\big(d(p_{i,t}, p_{(-i),t}), x_t\big)\\
&= \Sigma_{t=\hat{t}+1}^{T}(\bar{p}_{i,t} - c)d(\bar{p}_{i,t}, p_{(-i),t})\\
&= 0
\end{aligned}$$

Putting all three terms together, agent i's revenue from playing $\bar{p}^{(i)}$ is as great as, or greater than that from playing $p^{(i)}$. □

E Supplementary experiments

E.1 Comparison of means

Figure 7 compares the arithmetic, geometric and generalized means. We vary the generalized mean's parameter γ between 0 and 1, showing that it interpolates the arithmetic and generalized means, providing a balance between the former's ability to deal with negative values, and the latter's ability to weigh outcomes with supra-competitive total profits, but a disparity in profit gain between agents, as less collusive than symmetrical ones.

E.2 PPO behavior analysis

Like in Section 5.5, we analyze PPO's learned strategies from its behavior during an eval episode in Figure 8 and from its response surfaces. The evaluation episode shows that PPO has learned to collude over multiple episodes, with both agents starting off highly collusive and gradually undercutting each other, ending up at the collusive level at the end of the episode. This contrasts DQN's tendency to rise in collusion during the episode, before defecting toward the end. PPO does not seem to punish collusion strongly. Its reaction surface (Figure 9) suggests that it instead relies on a mutual understanding of collusion as a slow price war, undercutting if the opponent prices higher but preferring to reset the price after an opponent's deviation.

E.3 Choice of price-action grid

In the constrained setting, we define the available actions to be a discretized grid in the interval between (and extending slightly beyond) the *constrained* Nash equilibrium and monopolistic optimum prices. This interval is narrower than in the unconstrained case, as the Nash equilibrium price increases, while the monopolistic price level stays the same. In an episodic setting, agents are effectively unconstrained at the beginning of an episode, so by doing this we are restricting some of their ability to strategize. However, as Figure 10 shows, agents quickly learn to only price between the constrained competitive and collusive interval, suggesting that restricting the price grid does not cut off a relevant part of the strategy space.

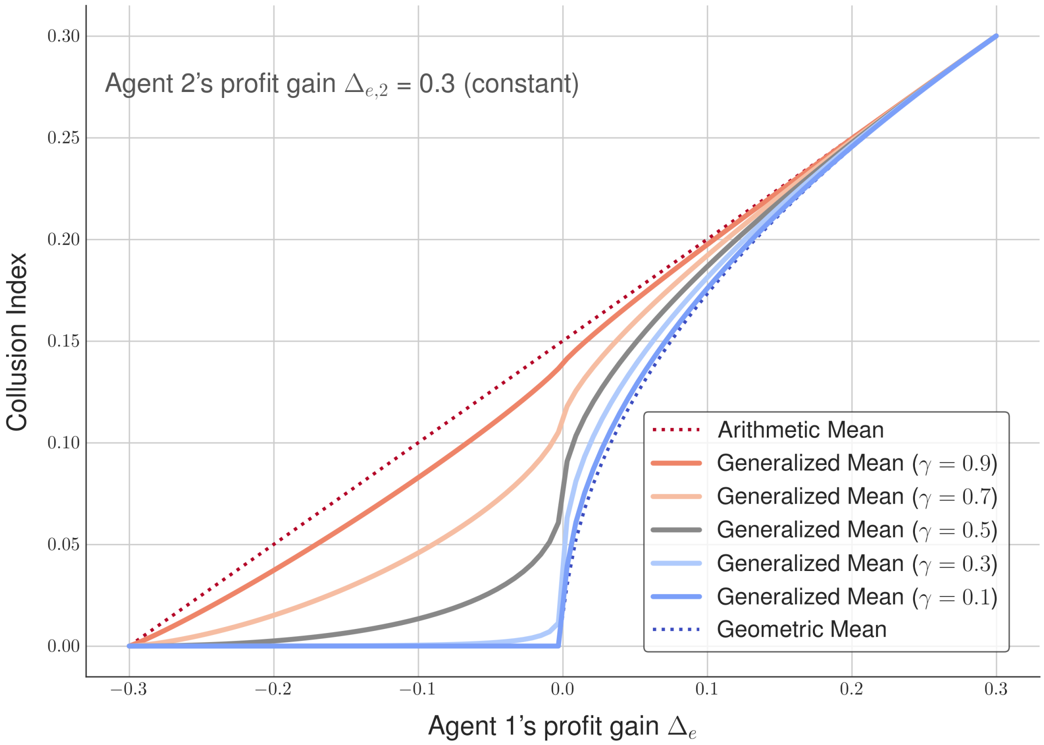


Figure 7: Showing how the generalized mean interpolates between the arithmetic and geometric means for choices of parameter $\gamma \in [0, 1]$, in the context of the collusion index measure.

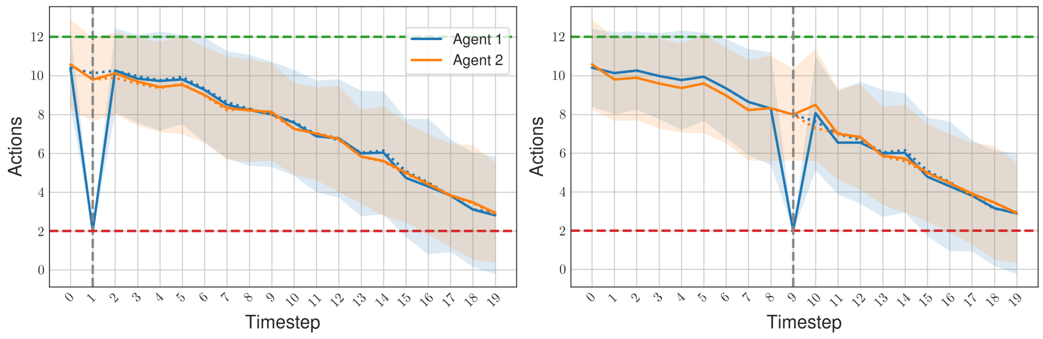


Figure 8: Behavior of two PPO agents during an episode after forcing one agent to deviate at time $t = 1$ and $t = 9$ respectively. Dotted lines indicate evolution without deviation. Deviations provoke a competitive reaction, with both agents quickly returning to collusion.

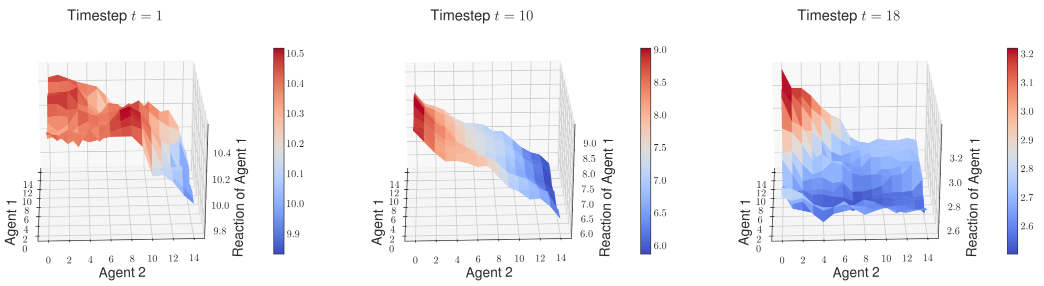


Figure 9: "PPO response surface". The surfaces show a PPO agent 1's learned response to a state given by both agent's prices (x- and y-axes), timestep and symmetric remaining inventory level.

E.4 Asymmetric inventory constraints

We have assumed symmetric inventory constraints so far. Illustrated in Figure 11, PPO agents with asymmetric inventory constraints of 440 and 400 per timestep respectively (compared to 440 for both agents in the main paper), obtain a collusion index of 0.44 compared to the 0.43 of the symmetric case. Both agents price collusively, although the chosen price levels now differ: the agent with the tighter inventory constraint chooses an even higher action than before. This suggests that small asymmetries do not impact the emergence of collusion.

E.5 Observability

So far, we have assumed full observability. To test that assumption's impact, we perform additional experiments. Figure 12 shows the evolution for DQN, qualitative results are identical for PPO.

First, we remove the opponent's inventory from an agent's observation. This has little impact on DQN's performance, with the collusion index dropping from 0.43 to 0.41 for PPO, and from 0.23 to 0.21 for DQN. Second, we do not allow agents to observe time within an episode. We see only a small impact on DQN's performance. The collusion index drops to 0.36 for PPO, and to 0.19 for DQN.

E.6 Geometric intuition on impact of inventory constraints on collusion

Figure 13 shows the normalized reward (profit) surfaces of agent 1 (green) and 2 (red) of the one-period game as functions of the prices both agents choose. We compare the surface under symmetric inventory capacities of different sizes, namely unconstrained (i.e., infinitely large inventory), lightly constrained (capacity 470) and strongly constrained (capacity 380). One can observe that the peak of each agent's reward surface lies on the opposite side of the diagonal through the price grid as their opponent's, namely on the side where they undercut their opponent's price and profit from capturing additional demand. Introducing an inventory constraint limits agents' profits from undercutting as they cannot capture the additional demand past their inventory limit. This pushes the peaks of both agents' reward surfaces closer to the diagonal, and each other. Further tightening the constraints while keeping the number of actions between the Nash and collusive optima equal and thus 'zooming in' on the price grid to an area near the peaks makes the peaks appear further apart. As mentioned in the main paper, we can imagine the duopoly dynamic as both learning agents trying to climb toward the peak of their respective reward surface. Collusion is achieved if agents climb the ridge along the diagonal. The closer the peaks are to each other and thus the monopolistic optimum on the diagonal, the smaller each agent's incentive to deviate and the smaller the negative impact of a deviation on their opponent, which eases cooperation. Tightening inventory constraints thus complicates the coordination problem, making collusion less likely.

E.7 Steering PPO toward competitive behavior

By increasing the noise in agents' learning targets, they can only learn best-response strategies, driving convergence toward competition. This is achieved by setting the "number of environments" parameter very high. PPO trains on a batch of data gathered from playing one or more parallel episodes against the same opponent, but with different random seeds. While our model has deterministic transitions, PPO has a stochastic policy, such that these episodes have different evolutions. With many different episodes in the learning batch, the PPO agents are likely not able to discern the opponent's underlying policy well and adapt to it, instead learning the (Nash) best response of competition. Figure 14a shows the quick initial convergence to competition that is then never deviated from. Figure 14b shows an evaluation episode of the trained learners, who play (imperfect) competition throughout the episode.

E.8 Unconstrained DQN learners

We analyze the setup of two DQN learners in an environment where inventory constraints are *not* simulated. Figure 15a shows the evolution of the training run. The learners show stronger collusion and keep increasing collusion throughout the entire run. This is different to the constrained setting, where collusion stops increasing once a stable level is hit (even if training time is extended).

Figure 15b shows the behavior of the trained agents during an evaluation episode. Unlike in the constrained setting, where collusion was built intra-episode, the unconstrained learners start the episode by already behaving collusively. Their response to forced deviation is stronger than in the constrained case, reacting more competitively. They display the same backward induction-type behavior, having learned that deviation toward the end of the episode is unable to be punished and therefore less risky.

E.9 Impact of agent deviation on episode profits

Figure 16 shows the evolution of two DQN agents' profits when forcing one agent to deviate. We observe that the deviating agent's temporary profit is tempered by the punishment in response, with both agents quickly returning to their normal strategy. The end-of-episode total profit is merely 0.2% lower than without the deviation, with the deviating agent only breaking even and the non-deviating agent taking a slight loss.

E.10 Additional hyperparameter comparisons

We show some additional plots of agent hyperparameter behavior in Figure 17 for PPO and Figure 18 for DQN.

Beginning with PPO, scaling the amount of initial entropy has a marginal effect on both convergence and collusiveness. The number of epochs, on the other hand, has a much bigger impact. More epochs of training per training step on the same batch of data allows PPO to fit their strategy to their opponent's much more effectively, increasing collusiveness while slightly reducing convergence. Lastly, increasing the number of minibatches and thus frequency of gradient updates helps PPO converge, but it does hurt collusiveness, which could be explained from the increased noise from smaller batches. A very low number of minibatches sees very stable training – perhaps too stable to effectively explore collusion.

DQN behaves similarly with regards to exploration and stable targets. A larger buffer size reduces convergence due to the increased variance in experiences that can be sampled (which are less up-to-date as buffer size increases). Larger buffers do help with establishing collusion, though, perhaps precisely because singular opponent deviations are less likely to be included in the next gradient step. We further observe that there is no strong dependence on initial exploration epsilon for either convergence or collusion. On the other hand, the interval between training episodes (as opposed to episodes where experience is gathered without a gradient update, i.e. only filling the replay buffer) does matter. Decreasing training frequency, and thus likelihood of immediate response to an opponent deviation increases collusive tendencies, but there is a limit – training too infrequently increases instability in the targets again and leaves DQN unable to react to positive exploratory moves.

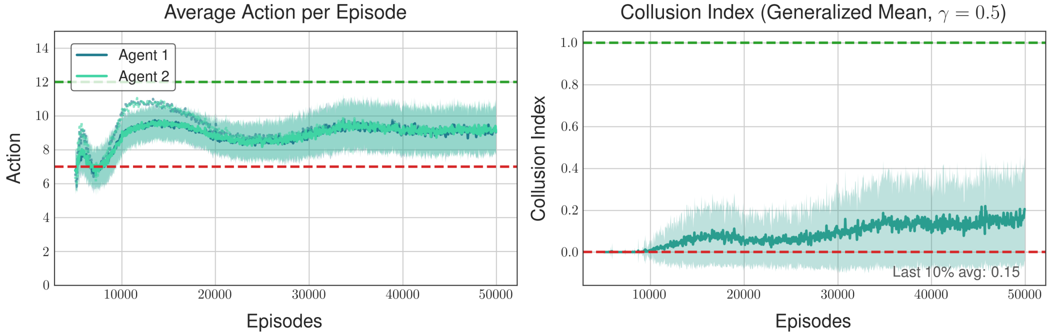


Figure 10: Choice of price-action grid: Training run evolution of two DQN learners using a price grid spanning the interval between the unconstrained Nash and monopolistic prices rather than the narrower constrained ones. Agents show the same pattern of learning competition and collusion.

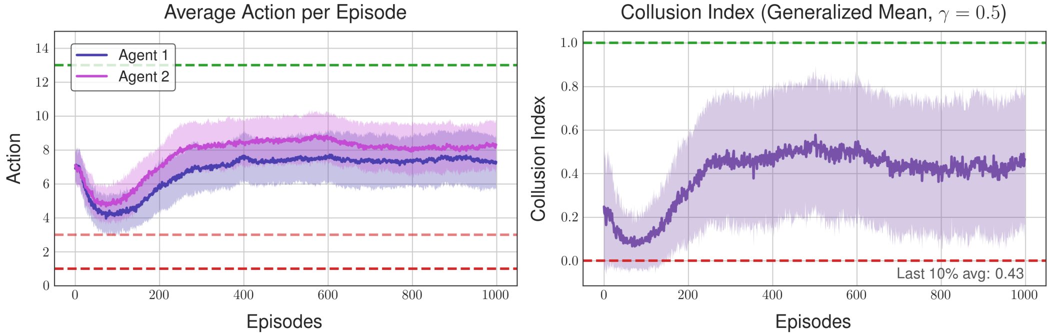


Figure 11: Asymmetric inventory constraints: Training run evolution of two PPO learners with asymmetric constraints of $440 \cdot T$ and $400 \cdot T$ respectively. Both agents settle at their now different collusive equilibrium price levels, but the overall collusion level remains the same.

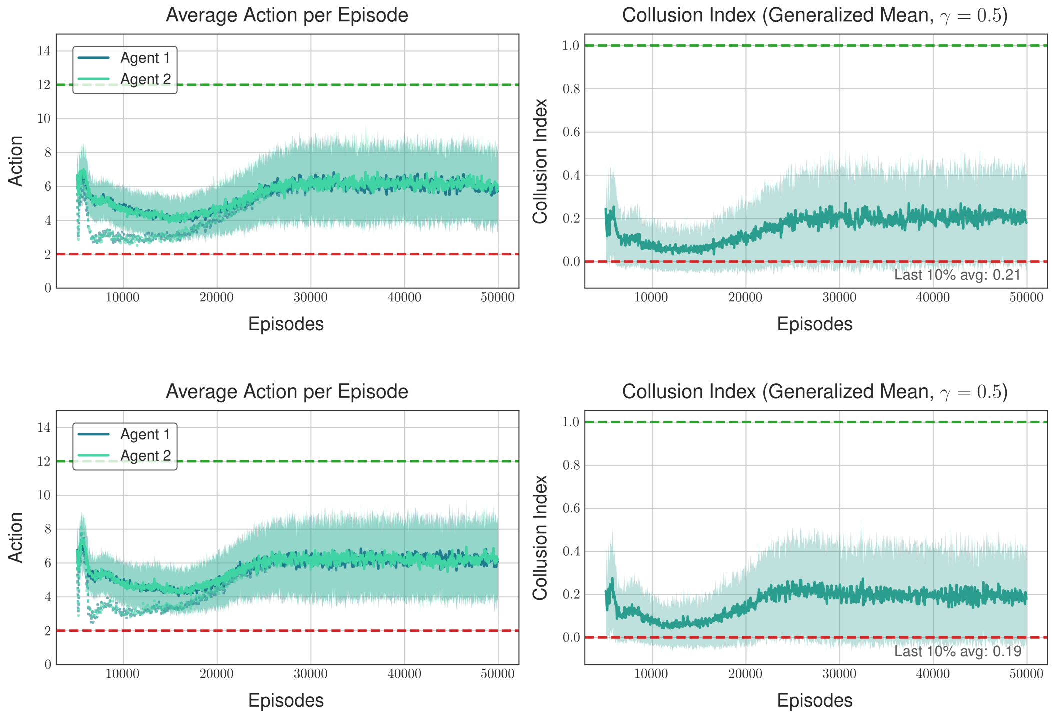
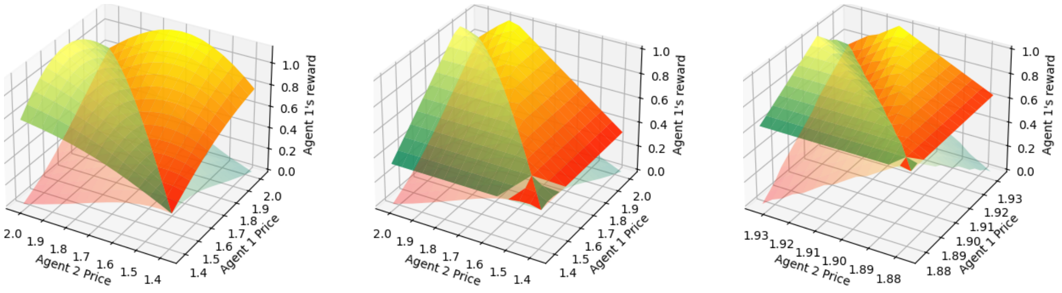


Figure 12: Observability: Evolution of training runs of two DQN learners that are prevented from observing their opponent's inventory (top) or the current timestep (bottom). We do not see a significant difference in behavior and collusive tendency.

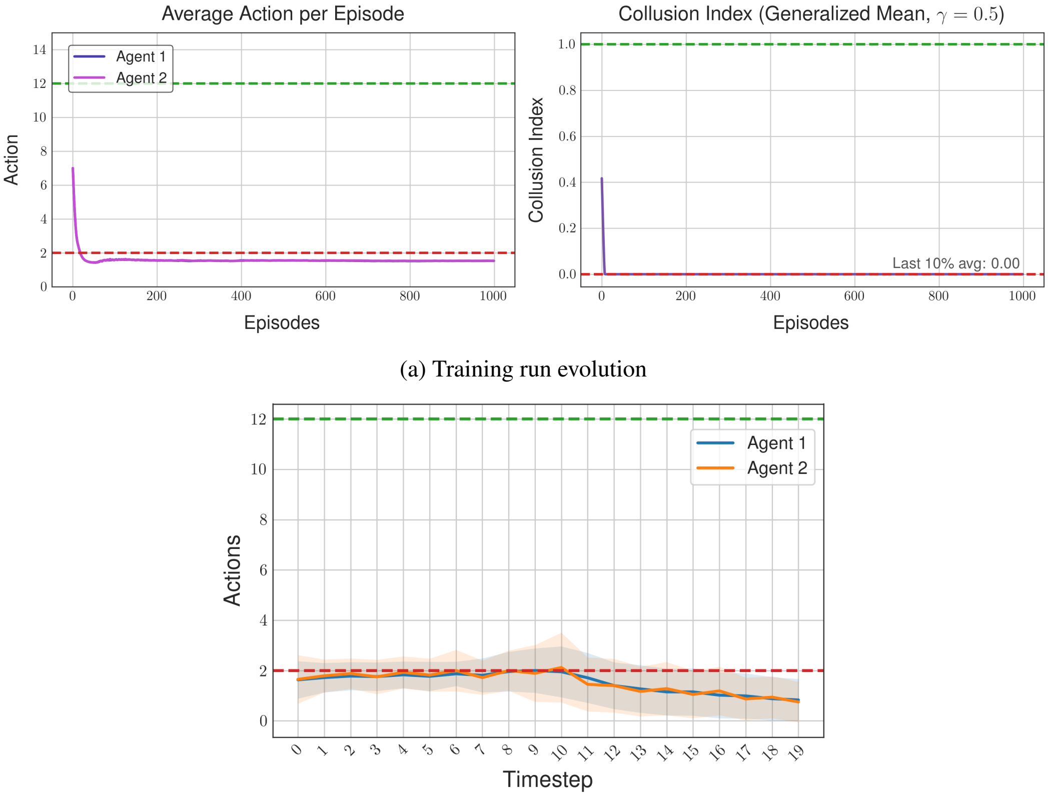


(a) Unconstrained inventory

(b) Light constraint

(c) Strong constraint

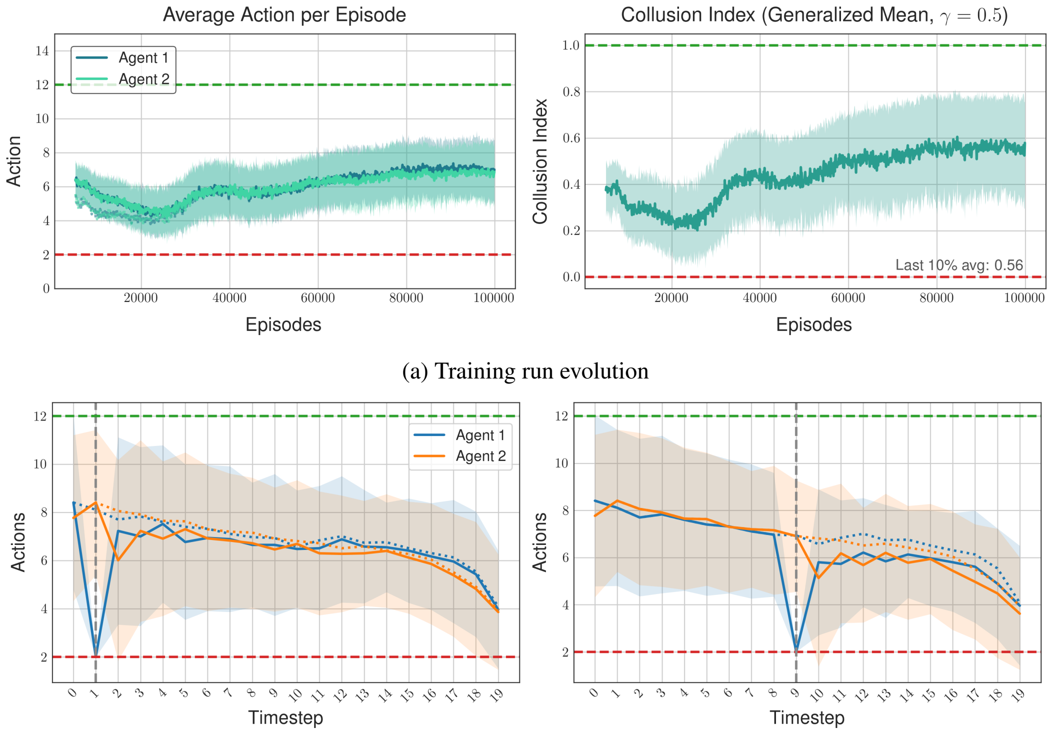
Figure 13: Geometric intuition on impact of inventory constraints on collusion: Normalized reward surfaces of agent 1 (green) and 2 (red) in a single period as functions of the prices both agents choose, with symmetric inventory capacities of different sizes (left to right: infinite, 470, 380). Introducing an inventory constraint pushes the peaks of both reward surfaces closer to each other, but tightening the constraints and thus 'zooming in' on the area near the peaks makes the peaks appear further apart, hindering collusion.



(a) Training run evolution

(b) Evolution within episode

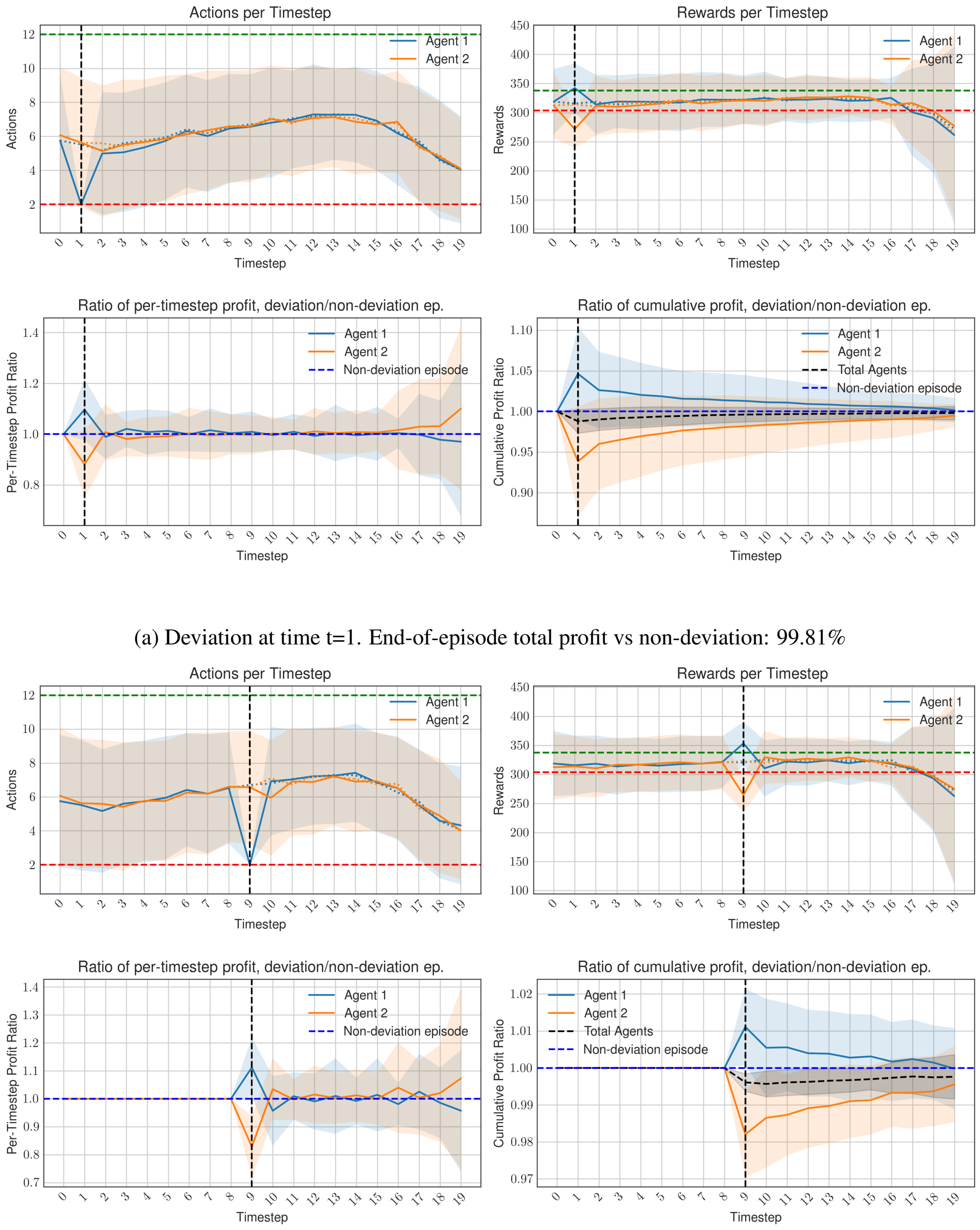
Figure 14: Competitive PPO: The evolution of a training run (a) and, once trained, within an episode (b) of two PPO learners trained with a very high "number of environments" parameter. They quickly converge to competition, and behave competitively throughout the entire episode.



(a) Training run evolution

(b) Forced deviation within episode

Figure 15: Unconstrained inventory: We train two DQN learners with an unconstrained inventory and plot the evolution of a training run (a) and the dynamics within an episode where one agent is forced to deviate to competition at time $t = 1$ and $t = 9$ (b). We observe stronger collusion than in the constrained case, and larger reactions to forced deviations.



(a) Deviation at time t=1. End-of-episode total profit vs non-deviation: 99.81%

(b) Deviation at time t=9. End-of-episode total profit vs non-deviation: 99.76%

Figure 16: “Profit impact of deviation”. Two trained DQN agents play an evaluation episode, with one agent being forced to deviate at different timesteps. We show the effect on individual and cumulative agent profit per single timestep, and over the entire evaluation episode. Total episode profit remains virtually unchanged by single deviations.

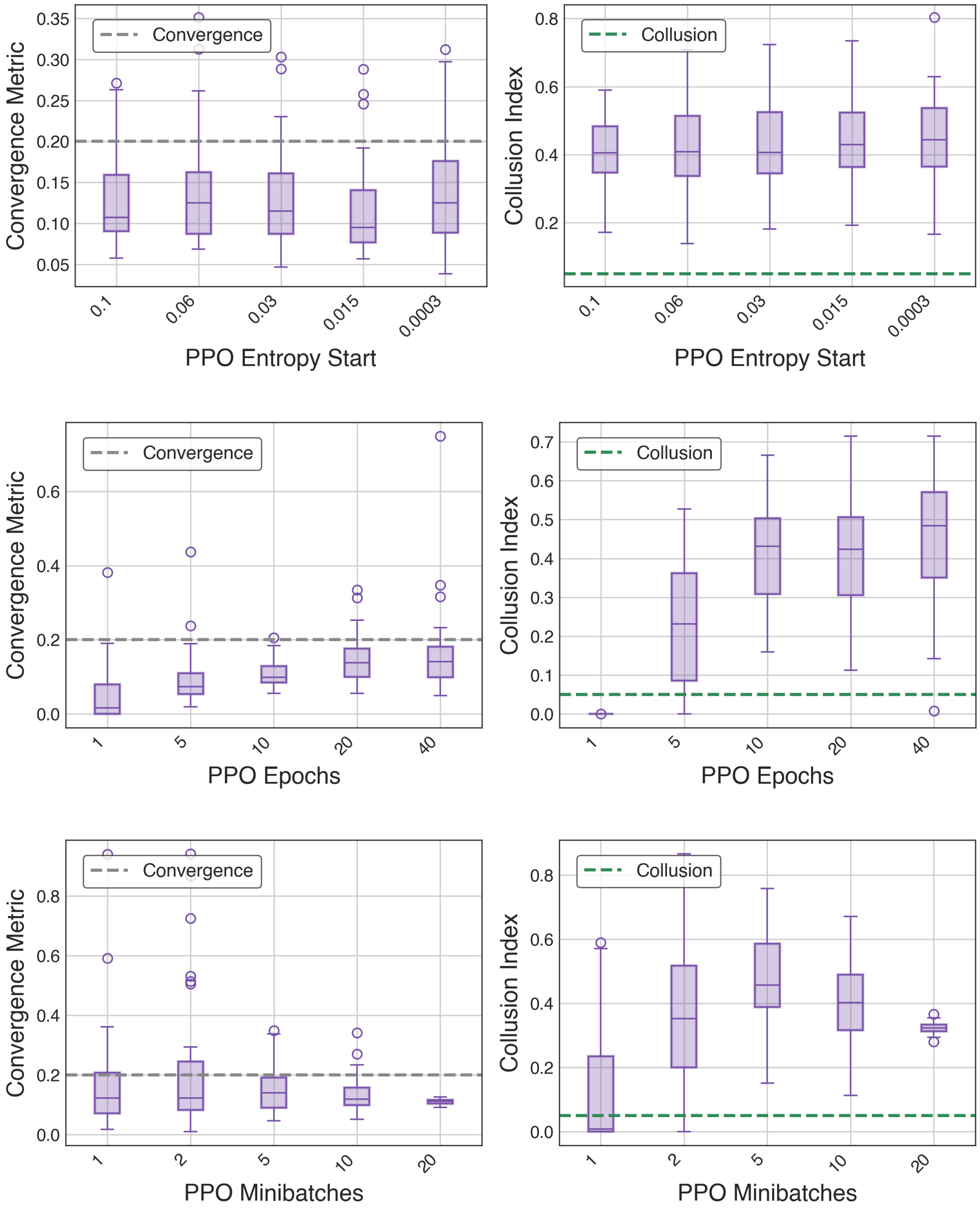


Figure 17: Convergence and collusion metrics for PPO training runs with varied starting entropy coefficient (top), number of epochs per training step (middle) and number of minibatches per training step (bottom). Collusion is robust against starting entropy and increases with more epochs or minibatches per training step.

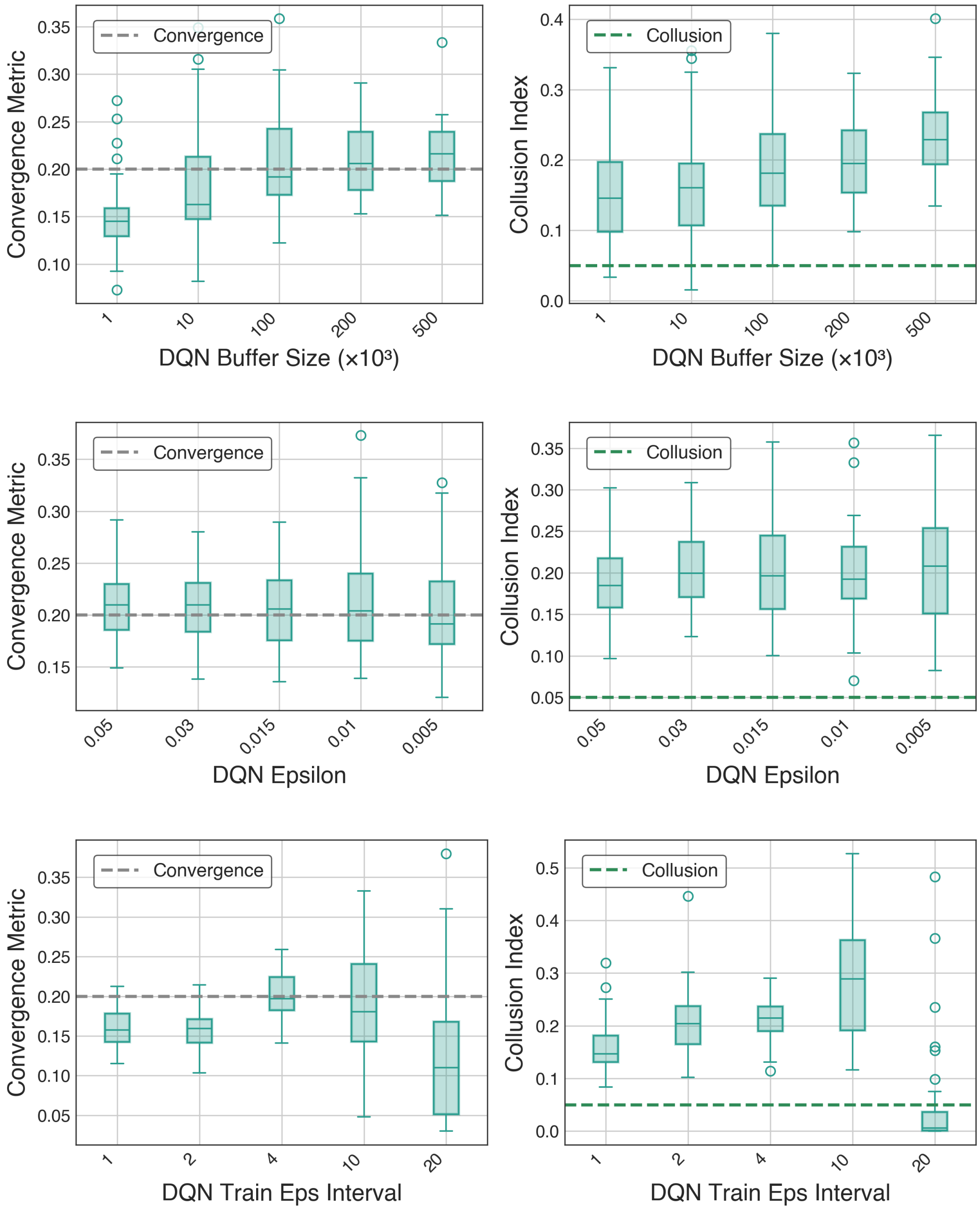


Figure 18: Convergence and collusion metrics for DQN training runs with varied buffer size in thousands (top), exploration epsilon (middle) and length of interval between training episodes (bottom). Larger buffer sizes increase collusion but reduce convergence, lower epsilon slightly worsens convergence without affecting collusion, and longer intervals improve convergence and collusion up to a point before becoming too sparse for learning.