Mitigating Retaliatory Algorithmic Collusion in Repeated Games

Karthik Sivachandran
Department of Computer Science
Purdue University
ksivacha@purdue.edu

Rohan Paleja
Department of Computer Science
Purdue University
rpaleja@purdue.edu

Abstract

Reinforcement learning agents trained to maximize their own reward in repeated interactions can converge to supra-competitive outcomes resembling explicit collusion, without communication or shared design. Existing mitigation approaches are largely tied to specific economic settings, like two-sided platforms and auctions, leaving open how to design interventions for general repeated games. We address this gap by formalizing the connection between empirical observations from prior work on Q-learning collusion and classical theory of Simple Penal Codes (SPCs). We show that an SPC with a collusive path and a distinct punishment path induces a quantifiable conditional dependence in agents' policies, detectable via the total variation distance between an agent's action distributions across cooperation and defection histories. Building on this connection, we propose CURB (Collusion Unwinding via Reward shaping and Belief injection), a reward-shaping framework that penalizes this Total Variation (TV) distance signal during Q-learning. For two-player games, we prove that, for sufficiently strong penalization, every equilibrium of the penalized objective that implements an SPC must have a stage-game Nash cooperative path. Thus, punishment threats cannot sustain a non-Nash collusive path at such equilibria. Empirically, CURB substantially reduces collusion by Q-learning agents in both Bertrand and Cournot Competition Repeated Games. We further demonstrate that CURB extends to deep Q-network agents in Bertrand competition, suggesting the mechanism has promise in generalizing beyond tabular Q-learning.

1 Introduction

In August 2024, the U.S. Department of Justice sued RealPage, alleging that its pricing software had helped competing landlords coordinate rent increases using competitively sensitive information shared with the platform, harming millions of renters [34]. The case highlights a broader concern: algorithmic pricing systems can facilitate coordination among competitors, potentially undermining competition and harming consumers.

A growing body of research has documented that reinforcement learning agents trained to maximize their own rewards in repeated interactions can independently converge to supra-competitive outcomes resembling collusion, without communication or shared design. The phenomenon has been observed in pricing oligopolies [12, 26], ad auctions [6], and richer multi-agent learning environments [27, 24]. Detecting and mitigating this behavior has become an active concern at the intersection of economics, computer science, and antitrust policy [10, 19, 11].

Despite the multi-domain nature of algorithmic collusion, existing mitigation frameworks are limited. Existing approaches often rely on setting-specific interventions, such as changing monitoring or auction formats, modifying platform rules, or auditing outcomes after the fact (Section 2).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We address a deeper question: what sustains the reward–punishment dynamics observed in algorithmic collusion across repeated games, and how can interventions provably dismantle it? We address this question by formalizing the connection between algorithmic collusion and the classical theory of repeated games. Our approach builds on a key observation from Calvano et al. [12]: Q-learning agents that converge to collusive outcomes implicitly implement *Simple Penal Codes (SPCs)* [1]—reward-punishment strategies in which deviations from collusion trigger retaliatory paths. Calvano et al. [12] noted this empirical resemblance, but the connection to Abreu's theory [1] has not been formalized or used for intervention. We make this bridge explicit to derive a mitigation framework. We make four contributions: **1)** We formalize the connection between Calvano's empirical observation and Abreu's SPC theory, proving that an SPC with a non-Nash cooperative path and a distinct punishment path induces quantifiable conditional dependence in agents' policies, detectable through a positive TV-distance lower bound, for two player games. **2)** For two-player games, we prove that, for sufficiently strong penalization, every equilibrium of the penalized objective that implements an SPC must have a stage-game Nash cooperative path. Thus, punishment threats cannot sustain a non-Nash collusive path at such equilibria. **3)** We propose CURB (Collusion Unwinding via Reward shaping and Belief injection), a heuristic implementation of this penalty. **4)** We empirically validate CURB across two repeated-game settings (Bertrand and Cournot competition) and two learning algorithms (tabular Q-learning and DQN), demonstrating that collusion is heavily mitigated.

For additional discussion of the theoretical scope, alternative mechanisms of algorithmic collusion, and practical considerations underlying CURB, we provide an FAQ in Appendix G, with links to relevant entries throughout the paper.

2 Related Work

Observation of algorithmic collusion. A substantial body of work has documented algorithmic collusion across diverse settings. Bertrand et al. [7] show that self-play Q-learners can collude in the iterated prisoner's dilemma, while Hansen et al. [18] and Asker et al. [3] demonstrate that even simple pricing algorithms can generate supra-competitive outcomes in standard pricing games. In oligopoly settings, Calvano et al. [12] and Klein [26] show that Q-learning agents can sustain collusion through reward–punishment dynamics in Bertrand competition, and Hettich [23] shows that deep Q-networks [30] converge to such outcomes more rapidly than tabular learners. Similar behavior has been observed in Cournot oligopoly [35], dealer markets [15], and auction environments [6]. More broadly, cooperation among learning agents arises naturally in sequential social dilemmas [27, 24]; while beneficial in some settings, these dynamics become problematic when coordination harms external parties. Recent work further shows that LLM-based agents can exhibit collusive behavior in strategic environments [17, 28]. Real-market evidence confirms these risks are not merely theoretical: Assad et al. [4] document an increase in duopoly margins where both stations adopt algorithmic pricing in Germany's retail gasoline market, with similar findings in online retailers of over-the-counter pharmaceuticals [9] and on e-commerce marketplaces [31].

Other work has proposed alternate mechanisms through which supra-competitive outcomes can arise without retaliatory threats, including a sequential no-regret-learning mechanism [2], whose classification as collusion is debated [20], and "spurious collusion," driven by insufficient exploration and optimistic initialization [14]. In contrast, CURB specifically targets punishment-sustained outcomes in simultaneous-move repeated games; these alternative mechanisms lie outside the scope of our guarantees (see Appendix G.1).

Mitigation approaches. Calvano et al. [13] demonstrate that imperfect monitoring of competitors' actions modestly weakens but does not eliminate collusion. Banchio and Skrzypacz [6] disrupt collusion in ad auctions by switching first-price to second-price, an intervention tied to auction format. Platform-side approaches, learned policies via Stackelberg POMDPs [8] and the platform pricing rules of Johnson et al. [25], require a centralized platform with authority to modify payoffs or rules and do not generalize across repeated games. Hartline et al. [21, 22] propose ex-post auditing, which detects collusion but does not prevent it.

Positioning of CURB. CURB targets the punishment-threat mechanism that sustains collusion rather than the conditions surrounding learning, the infrastructure around the agents, or the symptoms collusion produces in market outcomes. Where prior approaches detect collusion after it occurs, redesign the strategic environment, or impose platform-side rules, CURB modifies the agent's own learning rule with a penalty derived from the equilibrium structure of repeated games, providing a formal guarantee, for two-player games, that punishment-sustained collusion does not survive at

equilibria of the penalized objective the learning rule is designed to approximate. The mechanism requires only observing competitors' actions, which is standard in any repeated game with public actions, and applies across repeated games in the same independent-learner settings where collusion arises, without changing the strategic environment. The penalty operates on conditional action distributions rather than domain-specific quantities such as prices or quantities, allowing the same intervention to be instantiated across different repeated-game settings through an appropriate defection criterion.

3 Preliminaries

All theoretical results are stated for a general stage game. Let $G = (\{A_i\}_{i=1}^n, \{r_i\}_{i=1}^n)$, where $N = \{1, \ldots, n\}$ is the set of players, $\mathcal{A} = \times_{i \in N} \mathcal{A}_i$ is the joint action space with $\mathcal{A}_i$ finite for each player i, and $r = (r_1, \ldots, r_n)$ with $r_i : \mathcal{A} \to \mathbb{R}$ is the profile of payoff functions. We denote a joint action at time t by $q(t) = (q_1^t, \ldots, q_n^t) \in \mathcal{A}$, where q_i^t is the action taken by player i at time t. The *supergame* $G^\infty(\delta)$ is the infinitely repeated version of G with common discount factor $\delta \in (0, 1)$. A *pure strategy* for player i is a sequence of functions $\sigma_i = (\sigma_i(1), \sigma_i(2), \ldots)$ where $\sigma_i(1) \in \mathcal{A}_i$ and $\sigma_i(t) : \mathcal{A}^{t-1} \to \mathcal{A}_i$ for $t \geq 2$, mapping the history of all past joint actions to an action at period t. A *strategy profile* is $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Sigma \equiv \times_{i \in N} \Sigma_i$, where Σ_i is the strategy set of player i, i.e., the set of all pure strategies available to player i. A *path* $Q = \{q(t)\}_{t=1}^\infty \in \mathcal{A}^\infty$ is an infinite sequence of joint action profiles. Every strategy profile σ generates a unique path $Q(\sigma)$ defined inductively by $q(\sigma)(1) = \sigma(1)$ and $q(\sigma)(t) = \sigma(t)(q(\sigma)(1), \ldots, q(\sigma)(t-1))$. The *discounted payoff* to player i from path Q is defined by $v_i(Q) = \sum_{t=1}^\infty \delta^{t-1} r_i(q(t))$.

Assumption 1 (Bounded Payoffs)**.** *There exist $r_{\min}, r_{\max} \in \mathbb{R}$ such that $r_i(a) \in [r_{\min}, r_{\max}]$ for all $i \in N$, $a \in \mathcal{A}$.*

Assumption 2 (Finite Action Space)**.** *$\mathcal{A}_i$ is finite for all $i \in N$.*

Assumption 3 (Markov State)**.** *The state $s_t \in \mathcal{S}$ observed by agents at period t is a sufficient statistic of the payoff-relevant history. The state space $\mathcal{S}$ is finite.*

Theoretical regime. Our results assume:

- (T1) The cooperative path Q^0 is pure: $q_i^0(t) \in A_i$ is deterministic for all $i \in N$, $t \geq 1$.
- (T2) Learned policies π_j are stationary policies mapping $S \to \Delta(A_j)$, where $\Delta(A_j)$ is the set of probability distributions on A_j. This includes deterministic greedy and ε-greedy policies.
- (T3) The total variation (TV) distance in the shaped reward is computed over the exact conditional action distributions induced by π_j.

3.1 Simple Penal Codes

We recall the central equilibrium concept from Abreu [1]. For definitions of Nash Equilibrium and subgame perfect equilibrium we defer to Rubinstein [32].

Definition 1 (Simple Strategy Profile)**.** *Let $Q^0, Q^1, \ldots, Q^n \in \mathcal{A}^\infty$ be paths. The* simple strategy profile *$\sigma(Q^0, Q^1, \ldots, Q^n)$ specifies:*

1. *play Q^0 until some player deviates singly*
2. *if player j deviates singly from any ongoing path, switch to Q^j immediately*
3. *simultaneous deviations leave the ongoing path unchanged*

Definition 2 (Simple Penal Code)**.** *A simple strategy profile $\sigma(Q^0, Q^1, \ldots, Q^n)$ is a* Simple Penal Code *(SPC) if it is a subgame perfect equilibrium of $G^\infty(\delta)$.*

For the statement of the sustainability condition, we introduce two quantities. The *one-shot deviation gain* for player j at period t along path Q^i is given by Equation 1.

$$\Delta_j(q_j^*, q_{-j}^i(t)) = r_j(q_j^*, q_{-j}^i(t)) - r_j(q^i(t)) \tag{1}$$

In Equation 1, $q_j^* \in \arg\max_{a \in \mathcal{A}_j} r_j(a, q_{-j}^i(t))$ is player j's best deviation. The *continuation value* of player j from period $t+1$ onward along path Q is given by Equation 2

$$v_j(Q;\, t+1) = \sum_{s=1}^{\infty} \delta^s r_j(q(t+s)). \tag{2}$$

The following result characterizes when a simple strategy profile is an SPC.

Proposition 1 (Abreu [1])**.** *Under Assumptions 1-3, the simple strategy profile $\sigma(Q^0, Q^1, \ldots, Q^n)$ is a subgame perfect equilibrium if and only if for all $j \in N$, $i \in \{0, \ldots, n\}$, and $t \geq 1$:*

$$\underbrace{\Delta_j(q_j^*,\, q_{-j}^i(t))}_{\text{gain from deviating}} \;\leq\; \underbrace{v_j(Q^i;\, t+1) - v_j(Q^j)}_{\text{loss from punishment}}. \tag{3}$$

We call Equation 3 the *sustainability condition*. Collusion is sustained when deviation gains are outweighed by punishment losses.

Remark 1 (One-shot deviation)**.** *Under Assumptions 1–3, Proposition 1 is equivalent to the* one-shot deviation principle*: σ is a subgame-perfect equilibrium iff no agent can profitably deviate for a single period and then revert to σ at any subgame.*

Remark 2. *We consider agents that learn via tabular Q-learning [36]. Full details of the update rule are provided in Appendix A. In the repeated game $G^\infty(\delta)$, the state observed by agents at period t is the k-period action history $s_t = (q(t-1), q(t-2), \ldots, q(t-k)) \in \mathcal{A}^k$.*

3.2 Collusion as an Implicit Simple Penal Code

We use the term *collusion* to refer to outcomes where agents sustain supra-Nash payoffs at the expense of a third party whose welfare falls below its Nash level.

Definition 3 (Collusive Strategy Profile)**.** *A strategy profile σ is* collusive *if it implements a non-trivial Simple Penal Code and $W(Q^0) < W(Q^{\text{Nash}})$. $W(Q) = \sum_{t=1}^{\infty} \delta^{t-1} w(q(t))$ is the discounted third-party welfare under path Q, $w : A \to \mathbb{R}$ is the per-period third-party welfare function, and Q^{Nash} is the path generated by the stage-game Nash equilibrium.*

The trivial SPC $\sigma(Q^{\text{Nash}}, \ldots, Q^{\text{Nash}})$ is non-collusive since $Q^0 = Q^{\text{Nash}}$ implies $W(Q^0) = W(Q^{\text{Nash}})$.

Observation 1. *When Q-learning agents converge to a collusive outcome in the sense of Definition 3, their learned policies $(\pi_1, \ldots, \pi_n)$ implicitly implement a Simple Penal Code $\sigma(Q^0, Q^1, \ldots, Q^n)$, where Q^0 is the collusive path satisfying $W(Q^0) < W(Q^{\text{Nash}})$ and each Q^j is a punishment path encoding state-contingent retaliation following a unilateral deviation by player j.*

Observation 1 is supported empirically by Calvano et al. [12], who document that converged Q-learning policies in the Bertrand game exhibit reward-punishment behavior consistent with an implicit Simple Penal Code: agents cooperate on a supra-Nash path and retaliate following unilateral deviations. We treat Observation 1 as an empirically motivated modeling assumption rather than a formally established property.

Intervention rationale. Under Obs. 1, collusion sustained by punishment threats requires a non-trivial SPC. This motivates us to detect and dismantle the punishment structure that sustains collusive outcomes rather than target collusive outcomes directly. Removing punishment is necessary but not sufficient on its own: agents may still settle on a punishment-free, supra-Nash policy profile, such as one in which both firms persist at monopoly prices regardless of the opponent's actions. This motivates complementing reward shaping with belief injection.

4 Detecting Simple Penal Codes via Total Variation Distance

We now develop the first component: a method for detecting whether a learned policy implements a punishment path. We begin by formalizing what it means for a learned policy to implement such a path and prove TV distance between conditional policy distributions is necessary for detection.

4.1 Formalizing Punishment Strategies in Learned Policies

When agents collude, their learned policies implicitly implement a Simple Penal Code 1. The punishment paths $Q^1, \ldots, Q^n$ emerge implicitly. Agent j has learned a punishment strategy if its policy depends on whether agent i defected previously. Let $Q^0 \in \mathcal{A}^\infty$ denote the cooperative path and define the *defection partition* of $\mathcal{A}_i$ at period t as:

$$\mathcal{C}_i(Q^0, t) = \left\{ a \in \mathcal{A}_i : a = q_i^0(t) \right\}, \tag{4}$$

$$\mathcal{D}_i(Q^0, t) = \mathcal{A}_i \setminus \mathcal{C}_i(Q^0, t), \tag{5}$$

where $q_i^0(t)$ is agent i's prescribed action under Q^0 at period t. The set $\mathcal{C}_i(Q^0, t)$ contains actions consistent with the cooperative path and $\mathcal{D}_i(Q^0, t)$ contains all deviations from it.

However, Q^0 may not be observable during learning, and computing the Nash equilibrium, the natural alternative reference point, is PPAD-hard in general [16]. We therefore introduce a general *defection criterion* that subsumes both approaches and accommodates settings where neither Q^0 nor the Nash equilibrium is directly available.

Definition 4 (Defection Criterion)**.** *A* defection criterion *for agent i is a function $\phi_i : \mathcal{S} \times \mathcal{A}_i \to \{0, 1\}$ mapping a state-action pair (s, a_i) to a binary defection indicator. The induced partition at period t is:*

$$\mathcal{D}_i^{\phi}(t) = \left\{ a_i^t \in \mathcal{A}_i : \phi_i(s_t, a_i^t) = 1 \right\}, \tag{6}$$

$$\mathcal{C}_i^{\phi}(t) = \mathcal{A}_i \setminus \mathcal{D}_i^{\phi}(t). \tag{7}$$

Remark 3. *Three instantiations of Definition 4 are provided in Appendix B.*

Assumption 4 (α-Precise Defection Criterion)**.** *The defection criterion ϕ_i has* precision at least $1 - \alpha$ *on both sides: for all $t \geq 1$,*

$$P\big(a_i^{t-1} \in \mathcal{D}_i(Q^0, t) \;\big|\; \phi_i(s_t, a_i^t) = 1\big) \geq 1 - \alpha, \tag{8}$$

$$P\big(a_i^{t-1} \in \mathcal{C}_i(Q^0, t) \;\big|\; \phi_i(s_t, a_i^t) = 0\big) \geq 1 - \alpha, \tag{9}$$

where $\alpha \in [0, 1/2)$.

Definition 5 (Punishment Strategy)**.** *Let $(\mathcal{D}_i^{\phi}, \mathcal{C}_i^{\phi})$ be the partition induced by an α-precise defection criterion (Assumption 4), and let $\varepsilon > 0$. Under the theoretical regime, a stationary policy $\pi_j : S \to \Delta(A_j)$ is a $(\mathcal{D}_i^{\phi}, \mathcal{C}_i^{\phi}, \varepsilon)$-*punishment strategy *if*

$$\left\| \pi_j(\cdot \mid a_i^{t-1} \in \mathcal{D}_i^{\phi}) - \pi_j(\cdot \mid a_i^{t-1} \in \mathcal{C}_i^{\phi}) \right\|_{TV} > \varepsilon, \tag{10}$$

where $\|\mu - \nu\|_{TV} = \frac{1}{2} \sum_{a \in A_j} |\mu(a) - \nu(a)|$ is the total variation distance.

4.2 TV Distance as a Necessary Detector

Proposition 2 (TV Distance Detects Punishment)**.** *Let $G^{\infty}(\delta)$ be a two-player repeated game under the theoretical regime, satisfying Assumptions 1–3. Let $Q^0 \in A^{\infty}$ be a non-Nash pure path, and let π_j be a stationary policy for agent j with path-consistent defection partition $(\mathcal{D}_i(Q^0, t), \mathcal{C}_i(Q^0, t))$. If there exist paths $Q^i, Q^j \in A^{\infty}$ such that $\sigma(Q^0, Q^i, Q^j)$ is a non-trivial SPC of $G^{\infty}(\delta)$ with Q^i induced by π_j's post-deviation behavior, then*

$$\left\| \pi_j(\cdot \mid a_i^{t-1} \in \mathcal{D}_i(Q^0, t)) - \pi_j(\cdot \mid a_i^{t-1} \in \mathcal{C}_i(Q^0, t)) \right\|_{TV} \geq \varepsilon^*(\delta), \tag{11}$$

where

$$\varepsilon^*(\delta) := \frac{(1 - \delta)\Delta^*}{r_{\max} - r_{\min}} \in (0, 1], \qquad \Delta^* := v_i(Q^0; t^* + 1) - v_i(Q^i) > 0, \tag{12}$$

and t^* is the period at which the sustainability condition (3) *holds for agent i.*

Proof sketch. Since Q^0 is non-Nash, the sustainability condition binds at some t^* with $\Delta^* > 0$. Construct an intermediate path $\widetilde{Q}$ where agent i continues cooperating while agent j plays punishment actions. By subgame perfection, the true punishment path Q^i weakly dominates $\widetilde{Q}$. Hence the continuation-value gap Δ^* must arise from differences in j's conditional action distributions between cooperation and punishment phases. Expected payoff differences are bounded by total variation distance times the payoff range, yielding $\mathrm{TV} \geq (1 - \delta)\Delta^*/(r_{\max} - r_{\min})$. A full proof is provided in Appendix C.1.2.

Corollary 1. *If the TV distance above equals zero, then the induced path Q^i coincides with Q^0, and π_j cannot implement any non-trivial punishment path against agent i.*

A full proof is provided in Appendix C.1.3.

Observation 2. *Let π_j^{Nash} denote the* stationary *stage-game Nash policy: π_j^{Nash} plays a stage-game Nash action at every period, independent of history. Then*

$$\|\pi_j^{\text{Nash}}(\cdot \mid a_i^{t-1} \in \mathcal{D}_i^{\phi}) - \pi_j^{\text{Nash}}(\cdot \mid a_i^{t-1} \in \mathcal{C}_i^{\phi})\|_{TV} = 0 \qquad \textit{for all } i, j \in N.$$

Remark 4 (Non-stationary Nash equilibria)**.** *Observation 2 concerns the* stationary *stage-game Nash policy, which is what the trivial SPC $\sigma(Q^{\text{Nash}}, \ldots, Q^{\text{Nash}})$ uses. The repeated game admits additional subgame-perfect equilibria that are history-dependent (for example, grim trigger); these are not the subject of Observation 2.*

Corollary 2 (α-precise extension)**.** *Let π_j be a stationary policy for agent j consistent with the SPC structure of Definition 1: π_j's action distribution at period t depends on whether the SPC is in the cooperation phase or the punishment phase triggered by player i's deviation, and is characterized by two phase-conditional distributions*

$$\pi_j^C := \pi_j(\cdot \mid \textit{cooperation phase}), \qquad \pi_j^D := \pi_j(\cdot \mid \textit{punishment phase against } i). \tag{13}$$

Under Assumptions 1– 4 with $\alpha < \varepsilon^*(\delta)/2$, for any SPC $\sigma(Q_0, Q_i, Q_j)$ with Q_0 non-Nash and $Q_i \neq Q_0$,

$$\|\pi_j(\cdot \mid \phi_i = 1) - \pi_j(\cdot \mid \phi_i = 0)\|_{TV} \geq \varepsilon^*(\delta) - 2\alpha > 0. \tag{14}$$

A full proof is provided in Appendix C.1.4.

Proposition 2 establishes that under the theoretical regime, any nontrivial punishment path Q_i against player i induces a detectable TV distance between player j's conditional action distributions. Corollary 2 extends this bound to an α-precise defection criterion, showing that the detection signal degrades with classifier noise. Section 5 exploits this result to construct a reward penalty that fires when a collusive SPC is forming.

5 Suppressing Collusive SPCs via Reward Shaping

Proposition 2 establishes that any collusive SPC produces a detectable TV distance signal. We now show how to use this signal to dismantle the punishment structure sustaining collusion. Our intervention has two complementary components: *reward shaping*, which raises the cost of executing a punishment strategy and removes retaliation as a fixed-point property; and *belief injection*, which biases the dynamics among the surviving fixed points toward the competitive equilibrium. The first is analyzed for an idealized penalized objective (Theorem 1). Both are implemented as heuristics inside Q-learning and evaluated empirically in Section 6.

5.1 Reward Shaping Against Punishment Strategies

We define the shaped reward for agent j at period t as $r_j^{shaped}(s_t, a_t) = r_j(s_t, a_t) - \lambda \cdot \|\pi_j(\cdot \mid a_i^{t-1} \in \mathcal{D}_i^{\phi}) - \pi_j(\cdot \mid a_i^{t-1} \in \mathcal{C}_i^{\phi})\|_{TV}$. Here, $\lambda > 0$ is the penalty strength. The penalty is nonzero when agent j implements a punishment strategy. By Observation 2, the penalty is zero at Nash equilibrium, so Nash play is never penalized.

Proposition 3 (Conditional invariance and punishment paths)**.** *Under the theoretical regime and Assumption 4 with $\alpha < \varepsilon^*(\delta)/2$, if agent j's policy π_j satisfies $\|\pi_j(\cdot \mid \phi_i = 1) - \pi_j(\cdot \mid \phi_i = 0)\|_{TV} = 0$ then every SPC $\sigma(Q^0, Q^i, Q^j)$ consistent with π_j, Q^0 non-Nash, satisfies $Q^0 = Q^i$ and agent j does not punish agent i.*

A full proof is provided in Appendix C.2.

To analyze the effect of the penalty, let $V_j(\pi_j, \pi_{-j})$ denote agent j's expected discounted return under the original reward function, for a fixed initial-state distribution. We define the corresponding penalized objective as

$$\begin{aligned} J_j(\pi_j, \pi_{-j}) := V_j(\pi_j, \pi_{-j}) \\ - \frac{\lambda}{1-\delta} \left\| \pi_j(\cdot \mid a_i^{t-1} \in \mathcal{D}_i^{\phi}) - \pi_j(\cdot \mid a_i^{t-1} \in \mathcal{C}_i^{\phi}) \right\|_{\text{TV}}. \end{aligned} \tag{15}$$

The factor $1/(1-\delta)$ accounts for the accumulation of the stationary policy's TV penalty over the infinite horizon.

Proposition 4 (Penalty eliminates nontrivial punishment paths)**.** *Let $G^\infty(\delta)$ be a two-player repeated game under the theoretical regime, satisfying Assumptions 1– 4 with $\alpha < \epsilon^*(\delta)/2$.*

Suppose $(\pi_i^\star, \pi_j^\star)$ satisfies

$$\pi_k^\star \in \arg\max_{\pi_k : S \to \Delta(A_k)} J_k(\pi_k, \pi_{-k}^\star), \qquad k \in \{i, j\},$$

and implements an SPC $\sigma(Q_0, Q_i, Q_j)$ with Q_0 non-Nash.

For any

$$\lambda > \lambda^* := \frac{r_{\max} - r_{\min}}{\epsilon^*(\delta) - 2\alpha},$$

no such penalized equilibrium exists; i.e., there is no policy profile (π_i^*, π_j^*) that constitutes a penalized equilibrium and implements an SPC with a non-Nash cooperative path Q^0.

Proof sketch. Suppose for contradiction that a penalized equilibrium $(\pi_i^\star, \pi_j^\star)$ implements an SPC $\sigma(Q^0, Q^i, Q^j)$ with Q^0 non-Nash. By Proposition 2 and Corollary 2, the corresponding conditional TV distance satisfies $\mathrm{TV}_j^\star \geq \varepsilon^*(\delta) - 2\alpha$. Construct a state-independent policy $\widehat{\pi}_j$ such that $\widehat{\pi}_j(s) = \mu_j$ for all $s \in S$, where $\mu_j \in \Delta(A_j)$ is a fixed action distribution. This gives $\widehat{\mathrm{TV}}_j = 0$. The worst-case loss in unshaped discounted payoff from switching to $\widehat{\pi}_j$ is $(r_{\max} - r_{\min})/(1-\delta)$, while the penalty savings are at least $\lambda(\varepsilon^*(\delta) - 2\alpha)/(1-\delta)$. For $\lambda > \lambda^\star$, the penalty savings exceed this loss, so $J_j(\widehat{\pi}_j, \pi_i^\star) > J_j(\pi_j^\star, \pi_i^\star)$, contradicting the assumption of penalized equilibrium. A full proof is provided in Appendix C.3.

5.2 Theoretical Guarantees

We now state the theoretical result characterizing the SPCs that can be implemented by equilibria of the penalized objective.

Theorem 1 (Main Theorem)**.** *Under Assumptions 1–4 with $\alpha < \varepsilon^*(\delta)/2$, for any $\lambda > \lambda^*$, every penalized equilibrium that implements an SPC $\sigma(Q^0, Q^i, Q^j)$ must have Q^0 as a stage-game Nash path.*

Proof. Suppose, for contradiction, that a penalized equilibrium implements an SPC $\sigma(Q^0, Q^i, Q^j)$ with Q^0 non-Nash. By Proposition 4, no such penalized equilibrium exists when $\lambda > \lambda^*$. This is a contradiction. Therefore, Q^0 must be a stage-game Nash path. □

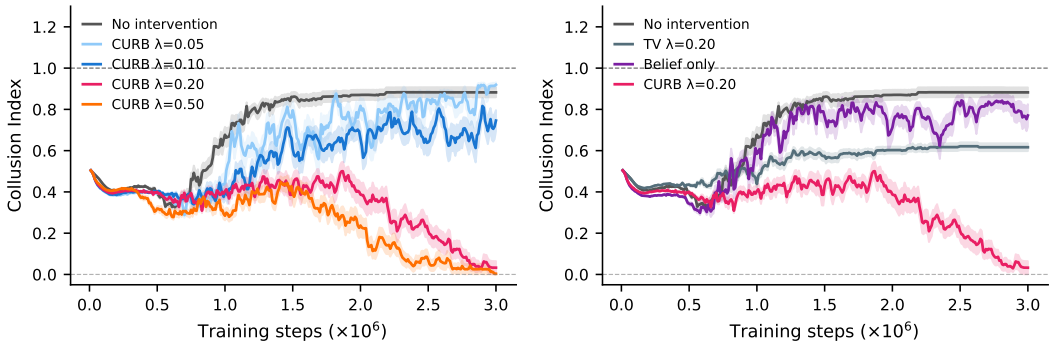
Theorem 1 establishes that sufficiently strong TV penalization rules out equilibria of the penalized objective that implement SPCs with non-Nash cooperative paths. In CURB, we use Q-learning as an approximate implementation of the penalized objective, with the corresponding learning updates described later in this section. We next establish that the stationary stage-game Nash equilibrium policy remains a fixed point under the shaped Q-learning updates.

Proposition 5 (Nash Stability)**.** *The stationary stage-game Nash equilibrium policy profile is a fixed point of shaped Q-learning.*

A full proof is provided in Appendix C.4.

Proposition 5 establishes that the stationary stage-game Nash equilibrium remains a fixed point of shaped Q-learning. However, this does not guarantee that the learning dynamics converge to Nash. Although reward shaping penalizes retaliatory behavior, an additional mechanism is needed to bias exploration toward competitive outcomes. We therefore complement reward shaping with belief injection, described next.

Belief Injection. The reward shaping component raises the cost of j having a policy conditioned on player i's action. This could lead to a situation where player j always chooses the monopoly price regardless of what i chooses. Normally, this would not be an issue as through the Q-learning updates player j will still reduce its price to stay competitive. However, an issue arises if they both settle at always playing the monopoly price. For this we complement this with a belief injection mechanism targeting agent i's incentive to cooperate. Collusion is sustained because agent i believes that defection will trigger punishment by agent j. If this belief is incorrect, if agent i expects no retaliation



(a) λ ablation

(b) Component ablation at λ=0.2

Figure 1: CURB on Bertrand Competition n=2

after defection, the cooperative path Q^0 becomes individually irrational and collusion fails. In every k steps we inject m synthetic experiences of the form $(s, a_i \in \mathcal{D}_i^{\phi}$, no retaliation by $j)$ into agent i's replay buffer, shifting its estimated Q-value for defection upward. Through this mechanism, we bias exploration during learning away from collusive policies. Related discussion in Appendix G.4.

The CURB Algorithm. At each stage game, every agent i selects an action and observes the reward, (ii) updates its Q-values using a TV-shaped reward that penalizes behavior conditional on past opponent defection, and (iii) periodically receives synthetic experiences depicting a unilateral deviation going unpunished. The penalty uses the maximum TV across opponents as a natural extension to $n > 2$ agents. The full algorithm is given in appendix D.

6 Experiments and Results

We evaluate CURB in Bertrand and Cournot repeated games under both tabular Q-learning and DQN. In Bertrand competition, agents choose prices each round; unilateral undercutting constitutes defection and price wars act as punishment. We use the Calvano et al. [12] setup with $K = 15$ actions and $n = 2$ unless otherwise stated. In Cournot competition, firms choose quantities under linear inverse demand; overproduction constitutes defection. We report the collusion index $\mathrm{CI} = (\bar{\pi} - \pi^{\mathrm{Nash}})/(\pi^{\mathrm{Mono}} - \pi^{\mathrm{Nash}})$ from [12], where $\mathrm{CI} = 0$ denotes Nash play and $\mathrm{CI} = 1$ full collusion. Since higher collusive profits in repeated oligopoly settings come at the expense of consumer surplus, increases in CI correspond to decreases in consumer welfare [33]. A summary of the experiments is in Appendix E.

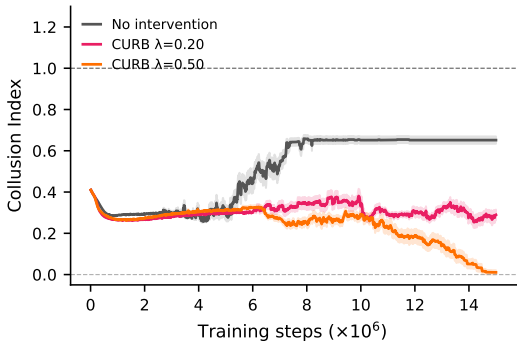
Defection criterion and belief injection. CURB requires a defection indicator ϕ_i to partition each agent's action history into "cooperative" and "defective" subsets. We use a simple online instantiation: an action is flagged as defective if it falls on the deviator side of the rolling-history median (below the median for Bertrand prices, above the median for Cournot quantities). Belief injection is implemented by periodically writing synthetic transitions into each agent's update target where that agent unilaterally defects against cooperative opponents and is not retaliated against, and is rewarded with the analytic profit at that synthetic joint outcome. In tabular settings the synthetic transition triggers a direct Q-update, in DQN it is pushed into the per-policy replay buffer.

Bertrand (tabular) ablations.

1. **Penalty strength λ sweep.** Figure 1(a) shows a clear threshold effect. $\lambda = 0.05$ behaves similarly to the no-intervention baseline, $\lambda = 0.10$ partially reduces collusion, and $\lambda \in \{0.2, 0.5\}$ drives CI close to Nash level.
2. **Component ablation.** Figure 1(b) compares TV shaping alone, belief injection alone, and full CURB at $\lambda = 0.2$. TV-only partially reduces collusion, belief-only remains near baseline, and only the combined intervention drives CI near zero.
3. **Platform-design baselines.** We compare CURB against Price-Directed Prominence (PDP) and Dynamic PDP [25], interventions in which a platform controls which seller is shown to consumers based on current price (PDP) as well as price history (DPDP). Because PDP and DPDP require a centralized platform while CURB modifies only the agents' update rule, we use a modified protocol: agents are trained under the platform rule, then evaluated in the full Bertrand environment, asking whether the platform-trained policy retains competitive behavior in the full

Condition	p_0	p_1	CI
No intervention	1.8142 ± 0.0142	1.8239 ± 0.0153	0.8824 ± 0.0245
PDP	1.4464 ± 0.0049	1.7198 ± 0.0290	0.0706 ± 0.0180
DPDP	1.5164 ± 0.0269	1.5671 ± 0.0302	0.0426 ± 0.0094
CURB λ=0.2	1.4729 ± 0.0000	1.4748 ± 0.0019	0.0029 ± 0.0028
CURB λ=0.5	1.4729 ± 0.0000	1.4748 ± 0.0019	0.0020 ± 0.0020

Table 1: Baseline comparison on Bertrand Competition n=2.



(a) Bertrand Competition n=3

λ	$\mathrm{CI}_{\mathrm{uni}}$	$\mathrm{CI}_{\mathrm{both}}$	$\Delta\bar{\pi}$
Control	0.883 ± 0.025	—	$+0.024 \pm 0.018$
0.2	0.572 ± 0.035	0.060 ± 0.025	$+0.030 \pm 0.035$
0.5	0.249 ± 0.049	0.024 ± 0.022	-0.040 ± 0.022
1	0.264 ± 0.050	0.002 ± 0.001	-0.028 ± 0.015
2	0.092 ± 0.039	0.002 ± 0.002	$+0.006 \pm 0.010$

(b) Unilateral adoption

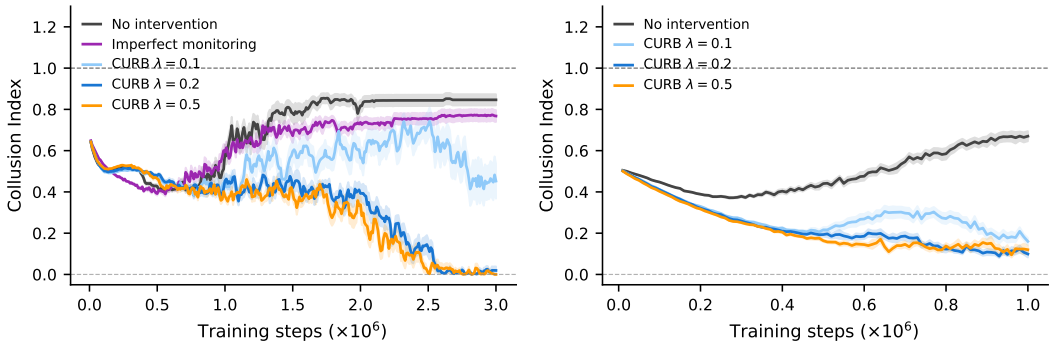
Figure 2: Left: CURB scales to $n = 3$ agents. Right: Unilateral adoption reduces collusion without evidence of systematic exploitation. $\mathrm{CI}_{\mathrm{uni}}$ and $\mathrm{CI}_{\mathrm{both}}$ denote the collusion index under unilateral and symmetric CURB adoption; $\Delta\bar{\pi} = \bar{\pi}_0 - \bar{\pi}_1$ is the normalized profit difference.

Bertrand game. Table 1 shows all three substantially reduce collusion relative to baseline, with CURB achieving the lowest CI. Unlike PDP/DPDP, which induce asymmetric pricing, CURB converges symmetrically near the Nash price. This indicates that CURB mitigates collusion in a competitively neutral way that does not advantage one seller over the other.

4. $n = 3$ **extension.** We test whether CURB scales beyond duopoly. Figure 2(a) shows the CI of the baseline reaches a substantially collusive equilibrium. CURB λ=0.2 partially reduces collusion. CURB $\lambda = 0.5$ drives the CI near zero for $n = 3$.
5. **Unilateral Adoption.** We test whether unilateral adoption of CURB reduces collusion and whether the adopting agent is exploited. In Bertrand $n = 2$, we apply full CURB to agent 0 and standard Q-learning to agent 1. We compare the Collusion Index against no intervention and symmetric CURB. To measure exploitation, we roll out the trained greedy policies from the monopoly state, discard the first 200 steps, and compare per-agent profits (normalized by monopoly profit) over the following 500 steps. Unilateral adoption substantially reduces collusion, bringing the CI from 0.883 to 0.092 at $\lambda = 2$. Meanwhile, the profit difference between the adopting and unmodified agents remains small and sign-inconsistent. This provides no evidence of systematic exploitation in this setting (Figure 2(b)).
6. **Additional experiments and ablations.** Further analyses, including discount-factor and exploration-decay baselines, a belief-injection hyperparameter sweep, Divergence ablation, forced-defection diagnostics, and TV-distance training trajectories are provided in Appendix F.

Cournot (tabular). We compare CURB against no intervention and the imperfect-monitoring modification of Calvano et al. [13], in which agents observe a noisy version of their opponents' quantities rather than the exact joint action. Figure 3(a) shows the CI over training for the no-intervention baseline, imperfect monitoring [13], and CURB at $\lambda \in \{0.1, 0.2, 0.5\}$. Imperfect monitoring partially reduces collusion, consistent with prior literature [13]. CURB λ=0.5 drives the CI to zero and outperforms imperfect monitoring. This demonstrates that CURB transfers across multiple repeated-game environments.

Bertrand (DQN) We replace tabular Q-learning with independent DQNs using 2-layer MLPs and replay buffers. Figure 3(b) shows that all CURB settings substantially reduce collusion relative to baseline, indicating the mechanism transfers beyond tabular learning.



(a) Cournot Competition n=2

(b) DQN-Bertrand Competition n=2

Figure 3: CURB transfers to Cournot competition and DQN.

7 Discussion & Conclusion

The defection criterion ϕ_i introduces a tunable parameter that, while requiring some domain knowledge about what constitutes cooperation and defection, also provides a useful degree of design flexibility. In many settings, certain joint action profiles may be cooperative yet welfare-neutral, while others are cooperative in appearance but harmful to third parties. Because ϕ_i partitions the action space explicitly, a designer can target specifically the subset of coordinated behaviors that reduce third-party welfare while grouping all remaining actions. This selectivity is a feature of the framework: rather than penalizing all history-dependent behavior indiscriminately, CURB can be tuned to intervene only against the coordination patterns that cause harm. A consumer-surplus-based instantiation illustrating this welfare-aware design choice is provided in Appendix G.2.

Limitations. Our formal guarantees apply only to two-player collusion; extending the analysis to n > 2 remains open. Belief injection is empirically effective but currently lacks formal guarantees. The framework targets collusion sustained by punishment dynamics, which prior literature has identified as a prominent mechanism by which algorithmic collusion arises. Whether CURB's detection signal would remain informative in settings where collusion may be sustained through alternative mechanisms is an open question. Our unilateral-adoption results show no evidence of systematic exploitation in the tested Bertrand setting, but do not establish that adoption is individually optimal for a payoff-maximizing firm; external incentives, such as regulation or deployment by an entity with a centralized welfare objective, may therefore be necessary. The primary advantage of CURB is its practicality in addressing already-deployed value-based RL when the external incentives are present. We also leave evaluation against rule-based and random opponents to future work. Our current detector focuses on immediate punishment and does not explicitly capture delayed punishment. A possible finite-horizon extension is discussed in Appendix G.3. Further, practical deployment of CURB also requires access to and the ability to modify the agents' learning processes, including their reward signals and replay buffers where applicable. This requirement may naturally be met in platforms that centrally make decisions for multiple participants, such as ride-sharing platforms using reinforcement learning for driver–rider matching [5]. CURB could be incorporated into such existing training pipelines to discourage punishment-sustained collusion.

Conclusion. We formalized the connection between algorithmic collusion and Simple Penal Codes, showing that non-trivial punishment strategies induce a detectable TV-distance signal in learned policies. Building on this connection, we proposed CURB, a framework that suppresses punishment-based collusion. Across Bertrand and Cournot competition under both tabular Q-learning and DQN, CURB consistently drove the Collusion Index near zero without modifying the game structure. As the risk of algorithmic collusion grows across markets and multi-agent systems, the need for principled interventions becomes increasingly important. By grounding mitigation in the mechanism of collusion rather than in the specifics of any one market or platform, CURB offers a step toward general-purpose tools for safeguarding competitive outcomes wherever autonomous agents learn to interact.

Acknowledgments and Disclosure of Funding

This research project was supported in part by the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of DARPA or the U.S. Government.

References

[1] Dilip Abreu. On the theory of infinitely repeated games with discounting. *Econometrica*, 56 (2):383–396, 1988. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1911077.

[2] Eshwar Ram Arunachaleswaran, Natalie Collina, Sampath Kannan, Aaron Roth, and Juba Ziani. Algorithmic Collusion Without Threats. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi: 10.4230/LIPIcs.ITCS.2025.10.

[3] John Asker, Chaim Fershtman, and Ariel Pakes. Artificial intelligence, algorithm design, and pricing. *AEA Papers and Proceedings*, 112:452–56, May 2022. doi: 10.1257/pandp.20221059. URL https://www.aeaweb.org/articles?id=10.1257/pandp.20221059.

[4] Stephanie Assad, Robert Clark, Daniel Ershov, and Lei Xu. Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy*, 132 (3):723–771, 2024. doi: 10.1086/726906. URL https://doi.org/10.1086/726906.

[5] Xabi Azagirre, Akshay Balwally, Guillaume Candeli, Nicholas Chamandy, Benjamin Han, Alona King, Hyungjun Lee, Martin Loncaric, Sébastien Martin, Vijay Narasiman, Zhiwei (Tony) Qin, Baptiste Richard, Sara Smoot, Sean Taylor, Garrett van Ryzin, Di Wu, Fei Yu, and Alex Zamoshchin. A better match for drivers and riders: Reinforcement learning at Lyft. *INFORMS Journal on Applied Analytics*, 54(1):71–83, 2024. doi: 10.1287/inte.2023.0083.

[6] Martino Banchio and Andrzej Skrzypacz. Artificial intelligence and auction design. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 30–31, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391504. doi: 10.1145/3490486.3538244. URL https://doi.org/10.1145/3490486.3538244.

[7] Quentin Bertrand, Juan Agustin Duque, Emilio Calvano, and Gauthier Gidel. Self-play q-learners can provably collude in the iterated prisoner's dilemma. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 3952–3975. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bertrand25a.html.

[8] Gianluca Brero, Eric Mibuari, Nicolas Lepore, and David Parkes. Learning to mitigate ai collusion on economic platforms. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 37892–37904. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/f746974abd33c0015ca583a267dac1fd-Paper-Conference.pdf.

[9] Zach Y. Brown and Alexander MacKay. Competition in pricing algorithms. *American Economic Journal: Microeconomics*, 15(2):109–56, May 2023. doi: 10.1257/mic.20210158. URL https://www.aeaweb.org/articles?id=10.1257/mic.20210158.

[10] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. Algorithmic pricing what implications for competition policy? *Review of Industrial Organization*, 55 (1):155–171, August 2019. ISSN 1573-7160. doi: 10.1007/s11151-019-09689-3. URL https://doi.org/10.1007/s11151-019-09689-3.

[11] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, Joseph E. Harrington, and Sergio Pastorello. Protecting consumers from collusive prices due to ai. *Science*, 370(6520):1040–1042, 2020. doi: 10.1126/science.abe3796. URL https://www.science.org/doi/abs/10.1126/science.abe3796.

[12] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. Artificial intelligence, algorithmic pricing, and collusion. *The American Economic Review*, 110(10):pp. 3267–3297, 2020. ISSN 00028282, 19447981. URL https://www.jstor.org/stable/26966472.

[13] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicoló, and Sergio Pastorello. Algorithmic collusion with imperfect monitoring. *International Journal of Industrial Organization*, 79:102712, 2021. ISSN 0167-7187. doi: https://doi.org/10.1016/j.ijindorg.2021.102712. URL https://www.sciencedirect.com/science/article/pii/S0167718721000059.

[14] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. Algorithmic collusion: Genuine or spurious? *International Journal of Industrial Organization*, 90:102973, 2023. ISSN 0167-7187. doi: https://doi.org/10.1016/j.ijindorg.2023.102973. URL https://www.sciencedirect.com/science/article/pii/S0167718723000541. The 49th Annual Conference of the European Association for Research in Industrial Economics, Vienna, 2022.

[15] Rama Cont and Wei Xiong. Dynamics of market making algorithms in dealer markets: Learning and tacit collusion. *Mathematical Finance*, 34(2):467–521, 2024. doi: https://doi.org/10.1111/mafi.12401. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mafi.12401.

[16] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 71–78, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595931341. doi: 10.1145/1132516.1132527. URL https://doi.org/10.1145/1132516.1132527.

[17] Sara Fish, Yannai A. Gonczarowski, and Ran I. Shorrer. Algorithmic collusion by large language models. arXiv preprint arXiv:2404.00806, 2024. URL https://arxiv.org/abs/2404.00806.

[18] Karsten T. Hansen, Kanishka Misra, and Mallesh M. Pai. Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms. *Marketing Science*, 40(1):1–12, January 2021. ISSN 1526-548X. doi: 10.1287/mksc.2020.1276. URL https://doi.org/10.1287/mksc.2020.1276.

[19] Joseph E Harrington. Developing competition law for collusion by autonomous artificial agents†. *Journal of Competition Law & Economics*, 14(3):331–363, 09 2018. ISSN 1744-6414. doi: 10.1093/joclec/nhy016. URL https://doi.org/10.1093/joclec/nhy016.

[20] Jason Hartline. Clarification of 'algorithmic collusion without threats', 2026. URL https://arxiv.org/abs/2602.22232.

[21] Jason D. Hartline, Sheng Long, and Chenhao Zhang. Regulation of algorithmic collusion. In *Proceedings of the 2024 Symposium on Computer Science and Law*, CSLAW '24, page 98–108, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703331. doi: 10.1145/3614407.3643706. URL https://doi.org/10.1145/3614407.3643706.

[22] Jason D. Hartline, Chang Wang, and Chenhao Zhang. Regulation of algorithmic collusion, refined: Testing pessimistic calibrated regret. In *Proceedings of the 2025 Symposium on Computer Science and Law*, CSLAW '25, page 108–120, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714214. doi: 10.1145/3709025.3712217. URL https://doi.org/10.1145/3709025.3712217.

[23] Matthias Hettich. Algorithmic collusion: Insights from deep learning. CQE Working Papers 9421, Center for Quantitative Economics (CQE), University of Muenster, Nov 2021. URL https://ideas.repec.org/p/cqe/wpaper/9421.html.

[24] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, Heather Roff, and Thore Graepel. Inequity aversion improves cooperation in intertemporal social dilemmas. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/7fea637fd6d02b8f0adf6f7dc36aed93-Paper.pdf.

[25] Justin P. Johnson, Andrew Rhodes, and Matthijs Wildenbeest. Platform design when sellers use pricing algorithms. *Econometrica*, 91(5):1841–1879, 2023. doi: https://doi.org/10.3982/ECTA19978. URL https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA19978.

[26] Timo Klein. Autonomous algorithmic collusion: Q-learning under sequential pricing. *The RAND Journal of Economics*, 52(3):538–558, 2021. ISSN 07416261, 17562171. URL http://www.jstor.org/stable/45419973.

[27] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, São Paulo, Brazil, 2017. URL https://www.ifaamas.org/AAMAS/aamas2017/proceedings/pdfs/p464.pdf.

[28] Ryan Y. Lin, Siddhartha Ojha, Kevin Cai, and Maxwell F. Chen. Strategic collusion of LLM agents: Market division in multi-commodity competitions. arXiv preprint arXiv:2410.00031, 2024. URL https://arxiv.org/abs/2410.00031.

[29] Yuhong Luo, Daniel Schoepflin, and Xintong Wang. Algorithmic collusion at test time: A meta-game design and evaluation, 2026. URL https://arxiv.org/abs/2602.17203.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. 2013. URL http://arxiv.org/abs/1312.5602. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.

[31] Leon Musolff. Algorithmic pricing facilitates tacit collusion: Evidence from e-commerce. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 32–33, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391504. doi: 10.1145/3490486.3538239. URL https://doi.org/10.1145/3490486.3538239.

[32] Ariel Rubinstein. Equilibrium in supergames with the overtaking criterion. *Journal of Economic Theory*, 21(1):1–9, 1979. ISSN 0022-0531. doi: https://doi.org/10.1016/0022-0531(79)90002-4. URL https://www.sciencedirect.com/science/article/pii/0022053179900024.

[33] Jean Tirole. *The Theory of Industrial Organization*, volume 1 of *MIT Press Books*. The MIT Press, 1 edition, December 1988. doi: None. URL https://ideas.repec.org/b/mtp/titles/0262200716.html.

[34] U.S. Department of Justice. Justice department sues RealPage for algorithmic pricing scheme that harms millions of American renters. Press release, Office of Public Affairs, August 2024.

[35] Ludo Waltman and Uzay Kaymak. Q-learning agents in a cournot oligopoly model. *Journal of Economic Dynamics and Control*, 32(10):3275–3293, 2008. ISSN 0165-1889. doi: https://doi.org/10.1016/j.jedc.2008.01.003. URL https://www.sciencedirect.com/science/article/pii/S0165188908000183.

[36] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992. doi: 10.1007/BF00992698. URL https://doi.org/10.1007/BF00992698.

A Q-Learning

Each agent i maintains a Q-function $Q_i : \mathcal{S} \times \mathcal{A}_i \to \mathbb{R}$ and updates it according to Equation 16.

$$Q_i(s_t, a_i^t) \leftarrow Q_i(s_t, a_i^t) + \alpha \left[r_i^t + \delta \max_{a' \in \mathcal{A}_i} Q_i(s_{t+1}, a') - Q_i(s_t, a_i^t) \right] \tag{16}$$

In Equation 16, $\alpha \in (0, 1)$ is the learning rate. The greedy policy derived from Q_i is $\pi_i(s) = \arg\max_{a \in \mathcal{A}_i} Q_i(s, a)$. During training agents act ε-greedily, selecting a random action with probability ε and the greedy action otherwise.

B Example Instantiations of the Defection Criterion

1. **Path-consistent** (when Q^0 is known):

   $$\phi_i(s, a) = \mathbf{1}\left[a \notin \mathcal{C}_i(Q^0, t)\right].$$

   Directly tied to Abreu's definition of deviation. Preferred when Q^0 is observable or estimable from a baseline run of standard Q-learning.

2. **Nash-deviation** (when stage game Nash payoffs are known):

   $$\phi_i(s, a) = \mathbf{1}\left[\max_{a_{-i} \in \mathcal{A}_{-i}} r_i(a, a_{-i}) \leq r_i^{Nash,G}\right],$$

   where $r_i^{Nash,G}$ denotes the Nash equilibrium payoff of the stage game G.

3. **Relative reward-percentile**:

   $$\phi_i(s_t, a_i^t) = \mathbf{1}\left[\rho_i^{t-1} \geq \hat{Q}_{1-p}\left(\rho_i^{(1:t-1)}\right)\right],$$

   where

   $$\rho_i^t = r_i^t - \frac{1}{n-1} \sum_{j \neq i} r_j^t$$

   is agent i's reward advantage over opponents at t, and $\hat{Q}_{1-p}(\rho_i^{(1:t)})$ is the online $(1-p)$-th percentile of observed relative advantages up to $t-1$. Agent i is classified as having defected when its reward advantage over opponents was unusually large, consistent with having captured excess surplus at opponents' expense. $p \in (0, 1)$ controls the fraction of periods classified as defections.

C Proofs

C.1 Proof of Proposition 2 and Corollaries

We first establish a lemma characterizing the optimality of punishment paths.

C.1.1 Punishment path is a best response

Lemma 1. *Let $\sigma(Q^0, Q^i, Q^j)$ be a subgame-perfect equilibrium of $G^\infty(\delta)$ under Assumptions 1–3. Then for any alternative path $\tilde{Q}$ generated by agent i playing an alternative strategy $\tilde{\sigma}_i$ against the other agents' prescribed post-deviation behavior,*

$$v_i(Q^i) \geq v_i(\tilde{Q}). \tag{17}$$

Proof: Consider the subgame beginning the period after agent i deviates singly from Q^0. By Definition 1, σ prescribes the continuation path Q^i with the other agents playing their Q^i-roles. By subgame perfection (Definition 2), σ_i restricted to this subgame is a best response to σ_{-i}. Since $\tilde{\sigma}_i$ is a feasible alternative and σ_i is optimal, $v_i(Q^i) \geq v_i(\tilde{Q})$.

C.1.2 Proof of Proposition 2

Define

$$\Delta_t^* := v_i(Q_0; t+1) - v_i(Q_i)$$

for each $t \geq 1$. Since Q_0 is not a stage-game Nash path, there exist a player i and a period t^* at which the prescribed action $q_i^0(t^*)$ is not a best response to $q_{-i}^0(t^*)$. Thus, the one-shot deviation gain satisfies

$$\Delta_i(q_i^{*,t^*}, q_{-i}^0(t^*)) > 0,$$

where

$$q_i^{*,t^*} \in \arg\max_{a \in A_i} r_i(a, q_{-i}^0(t^*)).$$

By the sustainability condition (3), applied to player i at period t^*,

$$\begin{aligned}\Delta_i(q_i^{*,t^*}, q_{-i}^0(t^*)) &\leq v_i(Q_0; t^*+1) - v_i(Q_i) \\ &= \Delta_{t^*}^*.\end{aligned}$$

Therefore, defining $\Delta^* := \Delta_{t^*}^*$, we obtain

$$\Delta^* > 0.$$

Define the intermediate path $\tilde{Q}$ on periods $t \geq t^*+1$ as follows:

- Agent i plays $q_i^0(t)$ (the cooperative on-path action);
- Agent j plays according to the punishment-phase distribution $\pi_j(\cdot \mid a_i^{t-1} \in D_i)$.

By construction, $\tilde{Q}$ is the path that results if agent i deviates at t^* – triggering the punishment phase – but then, contrary to optimality, continues to play the cooperative action $q_i^0(\cdot)$ while j punishes. This is a feasible alternative strategy for i in the post-deviation subgame, so Lemma 1 applies:

$$v_i(Q^i) \geq v_i(\tilde{Q}). \tag{18}$$

Decompose:

$$\Delta^* = v_i(Q^0; t^*+1) - v_i(Q^i) = \underbrace{\left[v_i(Q^0; t^*+1) - v_i(\tilde{Q}; t^*+1)\right]}_{\text{pure } j\text{-effect}} + \underbrace{\left[v_i(\tilde{Q}; t^*+1) - v_i(Q^i)\right]}_{\leq 0 \text{ by (18)}}. \tag{19}$$

Hence

$$v_i(Q^0; t^*+1) - v_i(\tilde{Q}; t^*+1) \geq \Delta^* > 0. \tag{20}$$

On path Q^0, agent i plays $q_i^0(\cdot)$ throughout, and agent j plays its cooperation-phase distribution $\pi_j(\cdot \mid C_i)$. On path $\tilde{Q}$, agent i still plays $q_i^0(\cdot)$ (by construction of $\tilde{Q}$), and agent j plays its punishment-phase distribution $\pi_j(\cdot \mid D_i)$. Thus Q^0 and $\tilde{Q}$ differ only in j's action distribution.

For each $s \geq 1$, define $h_s : A_j \to [r_{\min}, r_{\max}]$ by

$$h_s(a_j) := r_i(q_i^0(t^*+s), a_j).$$

Then

$$v_i(Q^0; t^*+1) - v_i(\tilde{Q}; t^*+1) = \sum_{s=1}^{\infty} \delta^{s-1} \Big(\mathbb{E}_{a_j \sim \pi_j(\cdot|C_i)}[h_s(a_j)] - \mathbb{E}_{a_j \sim \pi_j(\cdot|D_i)}[h_s(a_j)] \Big). \tag{21}$$

For any function $h : A_j \to [r_{\min}, r_{\max}]$ and any distributions μ, ν on A_j,

$$|\mathbb{E}_\mu[h] - \mathbb{E}_\nu[h]| \leq (r_{\max} - r_{\min}) \cdot \|\mu - \nu\|_{TV}. \tag{22}$$

(For any constant c, $\mathbb{E}_\mu[h] - \mathbb{E}_\nu[h] = \sum_a (h(a) - c)(\mu(a) - \nu(a))$. Choose $c = (r_{\max} + r_{\min})/2$ so $|h(a) - c| \leq (r_{\max} - r_{\min})/2$; then $|\mathbb{E}_\mu[h] - \mathbb{E}_\nu[h]| \leq \frac{r_{\max} - r_{\min}}{2} \sum_a |\mu(a) - \nu(a)| = (r_{\max} - r_{\min}) \|\mu - \nu\|_{TV}$.)

Let $\mathrm{TV} := \|\pi_j(\cdot|D_i) - \pi_j(\cdot|C_i)\|_{TV}$. Applying (22) termwise to (21) with $\mu = \pi_j(\cdot|C_i)$, $\nu = \pi_j(\cdot|D_i)$:

$$\left|v_i(Q^0; t^* + 1) - v_i(\tilde{Q}; t^* + 1)\right| \leq \sum_{s=1}^{\infty} \delta^{s-1} \cdot (r_{\max} - r_{\min}) \cdot \mathrm{TV} \tag{23}$$

$$= \frac{r_{\max} - r_{\min}}{1 - \delta} \cdot \mathrm{TV}. \tag{24}$$

Since the left-hand side of (24) is positive by (20), we may drop the absolute value and combine with (20):

$$\Delta^* \leq \frac{r_{\max} - r_{\min}}{1 - \delta} \cdot \mathrm{TV}. \tag{25}$$

$$\mathrm{TV} \geq \frac{(1 - \delta)\Delta^*}{r_{\max} - r_{\min}} = \varepsilon^*(\delta) > 0. \tag{26}$$

Since $\Delta^* > 0$ and the payoff range $r_{\max} - r_{\min}$ is finite by Assumption 1, $\varepsilon^*(\delta) \in (0, 1]$.

C.1.3 Proof of Corollary 1

Suppose

$$\left\|\pi_j(\cdot \mid a_i^{t-1} \in D_i(Q_0, t)) - \pi_j(\cdot \mid a_i^{t-1} \in C_i(Q_0, t))\right\|_{TV} = 0.$$

Since $\varepsilon^*(\delta) \in (0, 1]$, we have $0 < \varepsilon^*(\delta)$, so the hypothesis gives

$$\|\pi_j(\cdot \mid D_i(Q_0, t)) - \pi_j(\cdot \mid C_i(Q_0, t))\|_{TV} < \varepsilon^*(\delta).$$

By the contrapositive of Proposition 2, there exist no paths $Q_i, Q_j \in \mathcal{A}^\infty$ such that $\sigma(Q_0, Q_i, Q_j)$ is a non-trivial SPC of $G^\infty(\delta)$ with Q_i induced by π_j's post-deviation behavior. In particular, π_j cannot implement any non-trivial punishment path against agent i.

Because π_j is a *stationary* policy, its action distribution at any period t depends only on the conditioning event $a_i^{t-1} \in D_i(Q_0, t)$ or $a_i^{t-1} \in C_i(Q_0, t)$, not on t itself or on any other feature of the history. The TV distance vanishing means

$$\pi_j(\cdot \mid a_i^{t-1} \in D_i(Q_0, t)) = \pi_j(\cdot \mid a_i^{t-1} \in C_i(Q_0, t)) \quad \text{for all } t,$$

so π_j's action distribution is *identical* regardless of whether agent i has defected. Consequently, j's play following any history in $D_i(Q_0, t)$ is statistically indistinguishable from its play along the cooperative path itself, so j takes no action that depends on i's deviation. Since the induced path Q_i is generated entirely by best-response dynamics against π_j, and π_j exhibits no history-dependence between the cooperative and defection partitions, the induced continuation play cannot diverge from the on-path behavior under Q_0. Hence $Q_i = Q_0$.

C.1.4 Proof of Corollary 2

By Proposition 2 applied to the path-consistent partition, $\|\pi_j^D - \pi_j^C\|_{TV} \geq \varepsilon^*(\delta)$.

By the simple-strategy-profile structure (Definition 1), π_j's action distribution at period t depends only on the current phase, not on the specific classification $\phi_i(s_t, a_i^t)$ of the current action beyond its role in phase triggering. Formally, for any event E:

$$\pi_j(\cdot \mid E, \text{cooperation phase}) = \pi_j^C, \tag{27}$$

$$\pi_j(\cdot \mid E, \text{punishment phase}) = \pi_j^D. \tag{28}$$

Hence, by the law of total probability,

$$\pi_j(\cdot \mid \phi_i = 1) = P(\text{punishment phase} \mid \phi_i = 1) \cdot \pi_j^D + P(\text{cooperation phase} \mid \phi_i = 1) \cdot \pi_j^C \tag{29}$$

$$= (1 - w_D)\, \pi_j^D + w_D\, \pi_j^C, \tag{30}$$

where $w_D := P(\text{cooperation phase} \mid \phi_i = 1) \leq \alpha$ by Assumption 4. Similarly,

$$\pi_j(\cdot \mid \phi_i = 0) = (1 - w_C)\, \pi_j^C + w_C\, \pi_j^D, \qquad w_C \leq \alpha.$$

Bound each distance to the true phase-conditional:

$$\|\pi_j(\cdot \mid \phi_i = 1) - \pi_j^D\|_{TV} = w_D \cdot \|\pi_j^C - \pi_j^D\|_{TV} \leq w_D \leq \alpha, \tag{31}$$

$$\|\pi_j(\cdot \mid \phi_i = 0) - \pi_j^C\|_{TV} = w_C \cdot \|\pi_j^D - \pi_j^C\|_{TV} \leq w_C \leq \alpha. \tag{32}$$

(As $\|\pi_j^D - \pi_j^C\|_{TV} \leq 1$.)

By the triangle inequality:

$$\|\pi_j(\cdot \mid \phi_i = 1) - \pi_j(\cdot \mid \phi_i = 0)\|_{TV} \geq \|\pi_j^D - \pi_j^C\|_{TV} - \|\pi_j(\cdot \mid \phi_i = 1) - \pi_j^D\|_{TV} - \|\pi_j(\cdot \mid \phi_i = 0) - \pi_j^C\|_{TV} \tag{33}$$

$$\geq \varepsilon^*(\delta) - 2\alpha. \tag{34}$$

Since $\alpha < \varepsilon^*(\delta)/2$, the right-hand side is strictly positive.

C.2 Proof of Proposition 3

Suppose $\|\pi_j(\cdot \mid \phi_i = 1) - \pi_j(\cdot \mid \phi_i = 0)\|_{TV} = 0$, and suppose $\sigma(Q^0, Q^i, Q^j)$ is any SPC consistent with π_j.

By the phase-conditional mixture identity (proof of Corollary 2),

$$\pi_j(\cdot \mid \phi_i = 1) = (1 - w_D)\, \pi_j^D + w_D\, \pi_j^C, \tag{35}$$

$$\pi_j(\cdot \mid \phi_i = 0) = (1 - w_C)\, \pi_j^C + w_C\, \pi_j^D, \tag{36}$$

where $w_D, w_C \in [0, \alpha]$. Subtracting and taking TV distance:

$$\|\pi_j(\cdot \mid \phi_i = 1) - \pi_j(\cdot \mid \phi_i = 0)\|_{TV} = (1 - w_D - w_C) \cdot \|\pi_j^D - \pi_j^C\|_{TV}. \tag{37}$$

Since $w_D, w_C \leq \alpha < 1/2$, we have $1 - w_D - w_C > 0$. The hypothesis forces $\|\pi_j^D - \pi_j^C\|_{TV} = 0$.

Apply Proposition 2: $\|\pi_j^D - \pi_j^C\|_{TV} = 0$ implies $Q^i = Q^0$.

C.3 Proof of Proposition 4

Suppose, for contradiction, that (π_i^*, π_j^*) is a penalized equilibrium implementing an SPC $\sigma(Q^0, Q^i, Q^j)$ with Q^0 non-Nash.

Since Q^0 is non-Nash, there exist a player i and a period t^* at which player i has a strictly profitable one-shot deviation from Q^0. By the sustainability condition (3),

$$0 < \Delta_i(q_i^{*,t^*}, q_{-i}^0(t^*)) \leq v_i(Q^0; t^* + 1) - v_i(Q^i).$$

Thus, $Q^i \neq Q^0$.

By Corollary 2, we obtain

$$\left\|\pi_j^*(\cdot \mid \phi_i = 1) - \pi_j^*(\cdot \mid \phi_i = 0)\right\|_{TV} \geq \varepsilon^*(\delta) - 2\alpha > 0.$$

Let

$$TV_j^* := \left\|\pi_j^*(\cdot \mid \phi_i = 1) - \pi_j^*(\cdot \mid \phi_i = 0)\right\|_{TV}.$$

Now consider an alternative stationary policy $\widehat{\pi}_j$ that is independent of the state, i.e.,

$$\widehat{\pi}_j(s) = \mu_j \qquad \forall s \in S,$$

for some fixed distribution $\mu_j \in \Delta(A_j)$. By construction,

$$\|\widehat{\pi}_j(\cdot \mid \phi_i = 1) - \widehat{\pi}_j(\cdot \mid \phi_i = 0)\|_{TV} = 0.$$

Consider the change in agent j's penalized objective from switching to $\widehat{\pi}_j$, holding the opponent's policy fixed:

$$J_j(\widehat{\pi}_j, \pi_i^*) - J_j(\pi_j^*, \pi_i^*)$$

$$= V_j(\widehat{\pi}_j, \pi_i^*) - V_j(\pi_j^*, \pi_i^*) + \frac{\lambda}{1 - \delta} TV_j^*.$$

By Assumption 1, the discounted unshaped payoff of either policy lies in

$$\left[\frac{r_{\min}}{1-\delta}, \frac{r_{\max}}{1-\delta}\right].$$

Consequently,

$$V_j(\widehat{\pi}_j, \pi_i^*) - V_j(\pi_j^*, \pi_i^*) \geq -\frac{r_{\max} - r_{\min}}{1-\delta}.$$

Combining this bound with the lower bound on TV_j^* gives

$$\begin{aligned} &J_j(\widehat{\pi}_j, \pi_i^*) - J_j(\pi_j^*, \pi_i^*) \\ &\geq \frac{\lambda(\varepsilon^*(\delta) - 2\alpha) - (r_{\max} - r_{\min})}{1-\delta}. \end{aligned}$$

For

$$\lambda > \lambda^* := \frac{r_{\max} - r_{\min}}{\varepsilon^*(\delta) - 2\alpha},$$

the right-hand side is strictly positive. Thus, $\widehat{\pi}_j$ yields a strictly higher penalized objective than π_j^* against the same opponent, contradicting the assumption that (π_i^*, π_j^*) is a penalized equilibrium. Therefore, no such penalized equilibrium exists.

C.4 Proof of Proposition 5

Let $\pi^{\text{Nash}} = (\pi_1^{\text{Nash}}, \ldots, \pi_n^{\text{Nash}})$ denote the stationary stage-game Nash profile, where each π_j^{Nash} plays a stage-game Nash action independent of state. By Observation 2,

$$\left\|\pi_j^{\text{Nash}}(\cdot \mid a_i^{t-1} \in D_i^{\phi}) - \pi_j^{\text{Nash}}(\cdot \mid a_i^{t-1} \in C_i^{\phi})\right\|_{TV} = 0$$

for all $i, j \in N$. Therefore the TV penalty in the shaped reward is identically zero under π^{Nash}, and the shaped reward equals the unshaped reward: $r_j^{\text{shaped}}(s, a) = r_j(s, a)$ for all (s, a).

Because π^{Nash} is state-independent, the continuation value

$$V_j^{\text{Nash}}(s) = \sum_{t=0}^{\infty} \delta^t \, \mathbb{E}\big[r_j(\pi^{\text{Nash}})\big] = \frac{r_j(\pi^{\text{Nash}})}{1-\delta}$$

does not depend on s. Hence the Q-value of action a in state s satisfies

$$Q_j^{\text{Nash}}(s, a) = r_j\big(s, a, \pi_{-j}^{\text{Nash}}\big) + \delta \cdot V_j^{\text{Nash}},$$

and the only a-dependent term is the immediate reward. Maximizing over a reduces to the stage-game best-response problem:

$$\arg\max_{a \in \mathcal{A}_j} Q_j^{\text{Nash}}(s, a) = \arg\max_{a \in \mathcal{A}_j} r_j\big(s, a, \pi_{-j}^{\text{Nash}}\big).$$

Since π_j^{Nash} is by definition a stage-game best response to π_{-j}^{Nash}, we have $\pi_j^{\text{Nash}}(s) \in \arg\max_a Q_j^{\text{Nash}}(s, a)$ for all s. Hence π^{Nash} is a greedy-policy fixed point of shaped Q-learning.

D Algorithm

Algorithm 1 CURB

Require: penalty strength $\lambda > \lambda^*$, history length W, injection period k, injection size m, defection criterion ϕ_i

Initialise Q_i for all $i \in [n]$; sliding action history $\mathcal{H}$ of length W, initially empty
for $t = 1, 2, \ldots$ **do**
  Each agent i selects a_i^t ε-greedily from $Q_i(s_t, \cdot)$
  Observe rewards r_i^t and next state s_{t+1}
  Push joint action a^t into $\mathcal{H}$ (evicting the oldest entry if $|\mathcal{H}| = W$)
  for each agent $j \in [n]$ **do**
    For each $i \neq j$, classify each of i's actions in $\mathcal{H}$ via ϕ_i into a defective set $\mathcal{D}_i$ and a cooperative set $\mathcal{C}_i$
    Compute the lag-1 conditional TV between j's response distributions:

$$\mathrm{TV}_t^{ij} = \left\| \hat{\pi}_j\big(\cdot \,\big|\, a_i^{t-1} \in \mathcal{D}_i\big) - \hat{\pi}_j\big(\cdot \,\big|\, a_i^{t-1} \in \mathcal{C}_i\big) \right\|_{\mathrm{TV}}$$

    Apply shaped reward $\tilde{r}_j^t = r_j^t - \lambda \cdot \max_{i \neq j} \mathrm{TV}_t^{ij}$
    Update $Q_j(s_t, a_j^t) \leftarrow Q_j(s_t, a_j^t) + \alpha\big[\tilde{r}_j^t + \delta \max_{a'} Q_j(s_{t+1}, a') - Q_j(s_t, a_j^t)\big]$
  end for
  if $t \bmod k = 0$ **then**
    for each agent $j \in [n]$ **do**
      Inject m synthetic transitions into j's buffer in which j plays a defective action $a_j \in \mathcal{D}_j$ while opponents continue cooperating.
    end for
  end if
end for

E Experiments

This appendix documents the environments, algorithms, hyperparameters, and training scale used for the experiments reported in Section 6.

E.1 Environments

Bertrand competition. We use the symmetric Bertrand oligopoly with multinomial-logit demand from Calvano et al. [12]. Each agent i chooses a price p_i from a discrete grid of $K = 15$ values spanning the stage-game Nash and monopoly prices with a 10% markup at each end. Demand follows a logit form with quality $a_i = 2$, price sensitivity $\mu = 0.25$, and zero marginal cost. We use $n = 2$ for the main experiments and $n = 3$ for the scaling test.

Cournot competition. We use the deterministic Cournot oligopoly from Calvano et al. [13]. Each agent chooses a quantity q_i from a discrete grid of $K = 15$ values spanning the Nash quantity $q^{\text{Nash}} = a/(n+1)$ and monopoly quantity $q^{\text{Mono}} = a/(2n)$ with $a = 200$ and $b = 1$. Per-period profit is $\pi_i = (a - b\sum_j q_j) \cdot q_i$.

Cournot with imperfect monitoring. For comparison against the imperfect-monitoring deterrent of Calvano et al. [13], we additionally implement the Cournot variant in which the demand intercept d_t is i.i.d. uniform on $\{290, 310\}$ and agents observe only the realised market price (not opponent quantities). The state space is the previous-period price discretised into 37 levels.

State encoding. Across all environments the state s_t is the previous joint action $(a_1^{t-1}, \ldots, a_n^{t-1})$ (or the previous price under imperfect monitoring), giving $|\mathcal{S}| = K^n$ in the standard case.

Reward normalisation. All per-period profits are divided by the per-agent monopoly profit so collusion-index calculations and cross-environment λ values are directly comparable. The collusion index is CI $= (\bar{\pi} - \pi^{\text{Nash}})/(\pi^{\text{Mono}} - \pi^{\text{Nash}})$, with CI $= 0$ at Nash and CI $= 1$ at full collusion.

E.2 Algorithms

Tabular Q-learning. Each agent maintains a Q-table $Q_i \colon \mathcal{S} \times \mathcal{A}_i \to \mathbb{R}$ updated according to Equation 16. Following Calvano et al. [12], we initialise $Q_i(s, a) = \mathbb{E}_{a_{-i}}[\pi_i(a, a_{-i})]/(1 - \delta)$ averaged over opponent actions, which seeds the table with reasonable state-independent value estimates.

DQN. For the function-approximation experiments we replace the tabular Q-tables with independent multi-agent DQN: one PyTorch Q-network per agent, with no weight sharing. Each network is a 2-layer MLP with 32 hidden units per layer and ReLU activations, trained with Adam, MSE loss, a hard target-network copy every $C = 100$ steps, and a replay buffer of size 10^4. State observations are the previous-period prices, normalised to $[0, 1]$.

E.3 CURB components

Defection criterion. The TV detector requires a defection indicator ϕ_i to partition agent i's past actions into "cooperative" and "defective" subsets. We use the simple median-split: at each timestep, ϕ_i flags an action as defective if it falls on the deviator side of the rolling-history median (below the median for Bertrand prices; above the median for Cournot quantities). A punishment-direction filter further rejects spurious detections by requiring agent j's mean action when i is below the median to lie below j's mean action when i is above the median.

TV penalty. The TV detector operates on a rolling window of $W = 500$ past joint actions with lag-1 conditioning: agent j's current-period action is conditioned on agent i's *previous* action, capturing the temporal causality of punishment. The shaped reward is $\tilde{r}_j = r_j - \lambda \cdot \max_{i \neq j} \text{TV}_t^{ij}$, with λ swept over $\{0.05, 0.1, 0.2, 0.5\}$.

Belief injection. Every $K_{\text{belief}} = 100$ environment steps we inject $M_{\text{belief}} = 5$ synthetic transitions per agent into the agent's update target. Each injected transition models a unilateral deviation by j against opponents continuing on the cooperative path, with the analytic stage-game profit as the reward. In the tabular setting this is a direct Q-update; in the DQN setting it is a synthetic entry in the agent's replay buffer.

Synthetic transition construction. Each injected transition is constructed as follows: the injected agent's deviation is fixed at the stage-game Nash price or quantity, while all opponents remain at the monopoly price or quantity. The state paired with each transition is sampled uniformly from the full joint state space. The resulting next state places the injected agent at the Nash action and all opponents at the monopoly action. The synthetic reward is the analytic profit at this joint outcome.

E.4 Hyperparameters

Table 2 lists the hyperparameters used across all experiments. All values are recorded in the run_config.json of each completed run for reproducibility.

E.5 Penalty Strength Sweep

Table 3 summarizes the penalty strengths swept over for each experiment.

Parameter	Tabular Q	DQN
Learning rate α	0.15	10^{-3}
Discount δ	0.95	0.95
Exploration decay β	4×10^{-6} (n=2), 1.2×10^{-6} (n=3)	4×10^{-6}
Convergence threshold t_{stable}	10^5	—
Replay buffer size	—	10^4
Target-network update C	—	100
Hidden units (per MLP layer)	—	32
Batch size	—	32
Action-grid size K	15	15
TV rolling-window W	500	500
Belief-injection interval k	100	100
Belief-injection size m	5	5

Table 2: Hyperparameters used across experiments.

Experiment	**Penalty strengths λ**
Bertrand ($n = 2$), main ablation	$\{0.05, 0.10, 0.20, 0.50\}$
Bertrand ($n = 2$), component ablation	$\{0.20\}$
Bertrand ($n = 2$), platform-design baseline	$\{0.20, 0.50\}$
Bertrand ($n = 3$)	$\{0.20, 0.50\}$
Unilateral Adoption	$\{0.2, 0.5, 1, 2\}$
JSD divergence ablation (Appendix F.4)	$\{0.10, 0.20, 0.50\}$
Cournot ($n = 2$)	$\{0.10, 0.20, 0.50\}$
DQN-Bertrand	$\{0.10, 0.20, 0.50\}$
Belief Injection Hyperparameter Sweep (Appendix F.3)	$\{0.20\}$
Forced-defection diagnostic (Appendix F.5)	$\{0.50\}$
TV-distance training trajectories (Appendix F.6)	$\{0.05, 0.10, 0.20, 0.50\}$

Table 3: Penalty strengths used across experimental settings.

E.6 Training scale

Table 4 summarizes the conditions, seed counts, and training horizon for each experiment.

Experiment	Conditions	Seeds	Env steps
Bertrand n=2	10 (baseline + belief-only + 4 TV-only + 4 CURB)	20	3×10^6
Bertrand n=3	3 (baseline + 2 CURB)	20	1.5×10^7
DQN-Bertrand	4 (baseline + 3 CURB)	20	10^6
Cournot n=2	5 (baseline + IM + 3 CURB)	20	3×10^6
Platform-design	5 (baseline + PDP + DPDP + 2 CURB)	20	3×10^6
Unilateral Adoption	9 (baseline + 4 unilateral + 4 symmetric CURB)	20	3×10^6
JSD ablation	10 (baseline + 3 TV + 3 JSD-norm + 3 JSD-raw)	20	3×10^6
Discount factor	5 ($\delta \in \{0.1, 0.25, 0.5, 0.75, 0.99\}$)	20	3×10^6
Exploration decay	4 ($\beta \in \{4, 10, 40, 100\} \times 10^{-6}$)	20	3×10^6
BI Hyperparameter Sweep	42 ($k \times m$ grid)	10	3×10^6
Forced defection	2 (baseline + TV-only)	20	3×10^6
TV-distance training trajectories	5 (baseline + CURB, $\lambda \in \{0.05, 0.1, 0.2, 0.5\}$)	20	3×10^6

Table 4: Training scale per experiment. "IM" denotes imperfect monitoring; "PDP" / "DPDP" denote the Price-Directed Prominence baselines.

E.7 Compute resources

Tabular Q-learning experiments run entirely on CPU; we use Python multiprocessing to run seeds in parallel (typically 32 workers per node). A complete Bertrand n=2 sweep (200 seed-condition pairs at 3×10^6 steps each) finishes in roughly 2 hours on a 32-core node. The Bertrand n=3 sweep, with

1.5×10^7 steps per seed, takes approximately 6–8 hours on the same hardware. The DQN-Bertrand sweep uses a single GPU (A100) per seed and completes in under 4 hours per node.

E.8 Reporting

All plotted curves show the mean across seeds with ±1 SEM bands. All tables report the mean $\pm$ SEM across seeds. We report final-CI metrics by averaging the per-step CI over the last 10% of training steps to reduce sensitivity to oscillations near convergence.

F Supplementary Results

F.1 Discount Factor Baseline

We evaluate standard (unshaped) tabular Q-learning in Bertrand duopoly while varying the discount factor δ, holding the remaining experimental settings at their baseline values. Following Calvano et al. [12], Q-values are initialized to the expected profit against a random opponent divided by $(1-\delta)$. Consequently, the initialization scale also varies with δ. Each condition is run for 3×10^6 environment steps across 20 seeds.

Discount factor δ	Collusion Index (CI)
0.10	0.2010 ± 0.0257
0.25	0.2046 ± 0.0286
0.50	0.2366 ± 0.0276
0.75	0.4028 ± 0.0390
$0.95^\dagger$	0.8828 ± 0.0245
0.99	0.9248 ± 0.0135

Table 5: Collusion index under different discount factors. The $\delta = 0.95$ reference value is taken from the baseline condition of the exploration-decay sweep, not a separate run in the discount-factor sweep.

Reducing the discount factor decreases collusion, but with diminishing returns. The Collusion Index falls to approximately 0.20–0.24 for $\delta \leq 0.5$, but does not approach the competitive outcome even at $\delta = 0.1$. This illustrates the limitations of discount factor reduction as a mitigation strategy: it reduces agents' forward-looking incentives generally rather than selectively targeting retaliatory behavior.

F.2 Exploration Decay Baseline

We evaluate standard (unshaped) tabular Q-learning in Bertrand duopoly while varying the exploration decay rate β, holding the remaining experimental settings at their baseline values. Exploration follows $\epsilon_t = e^{-\beta t}$, with the discount factor fixed at $\delta = 0.95$. Each condition is run for 3×10^6 environment steps across 20 seeds.

Exploration decay β	Collusion Index (CI)
4×10^{-6}	0.8828 ± 0.0245
1×10^{-5}	0.7909 ± 0.0321
4×10^{-5}	0.7495 ± 0.0105
1×10^{-4}	0.7041 ± 0.0038

Table 6: Collusion index under different exploration decay rates.

Faster exploration decay reduces collusion only modestly: the CI decreases from 0.88 at the baseline decay rate to 0.70 at the highest rate tested, remaining substantially above the competitive level. Thus, accelerating exploration decay alone does not eliminate collusion within the range tested.

F.3 Belief Injection Hyperparameter Sweep

We evaluate the sensitivity of the full CURB algorithm (TV shaping and belief injection) in Bertrand duopoly to its two belief-injection hyperparameters: the injection period k and the number of synthetic transitions per injection m. We sweep the full grid $k \in \{10, 25, 50, 100, 250, 500, 1000\}$ and $m \in \{1, 2, 5, 10, 25, 50\}$, holding all other settings at their main-experiment values, including the TV penalty strength $\lambda = 0.2$. Each condition is run for 3×10^6 environment steps across 10 seeds.

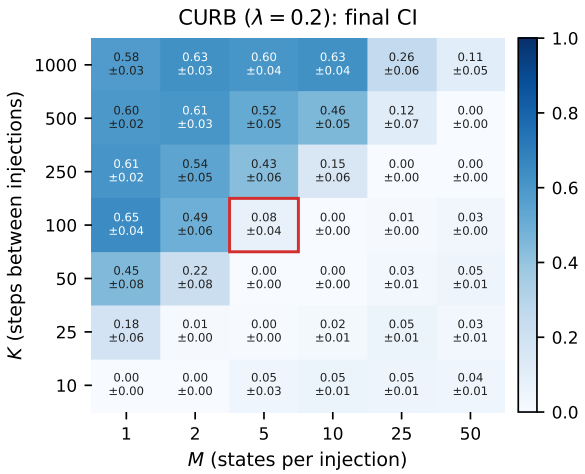


Figure 4: Final mean collusion index of CURB ($\lambda = 0.2$) across belief-injection period k and injection size m. The outlined cell marks the default setting ($k = 100$, $m = 5$).

The outcomes shown in figure 4 are affected primarily by the injection rate m/k, the number of synthetic updates per environment step. For every configuration with $m/k \geq 0.1$, CURB reduces the CI to at most 0.05. When injection is sparse, the CI remains between 0.43 and 0.65, indicating that belief injection is then too infrequent to affect learning. Overall, CURB does not depend on a finely tuned choice of k and m: any injection rate of roughly one synthetic update per ten environment steps or more drives collusion close to the competitive level.

F.4 Divergence Ablation

We chose TV distance for CURB due to its simplicity. However, from an information theoretic perspective it is natural to ask whether the TV penalty can be replaced by alternative divergence measures. We compare TV against Jensen–Shannon Divergence (JSD) at matched λ, using two conventions: JSD-norm rescales to $[0, 1]$ to match TV's range, JSD-raw keeps its native range $[0, \ln 2]$.

Figure 5 shows that all three metrics qualitatively reduce collusion below baseline, but at any given λ, TV $>$ JSD-norm $>$ JSD-raw in mitigation strength.

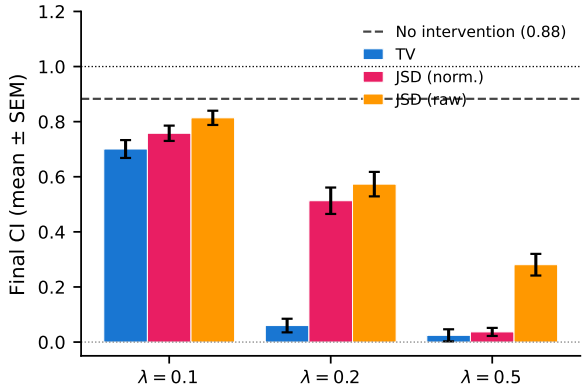


Figure 5: TV vs JSD divergence ablation, n=2.

F.5 Forced-defection test

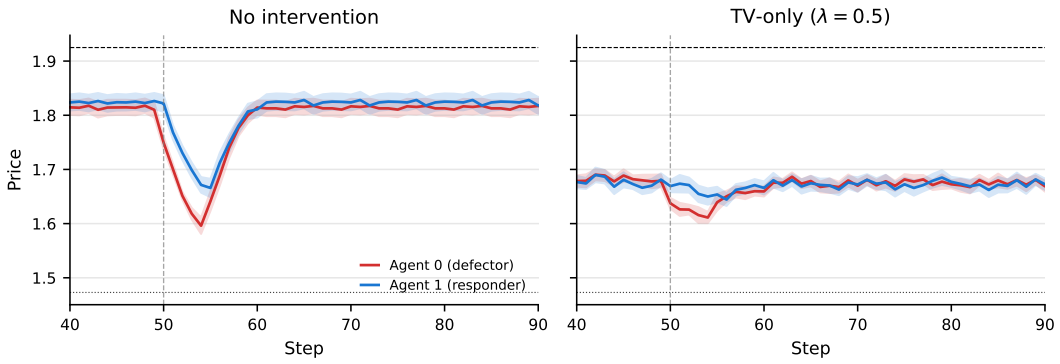


Figure 6: Forced-defection test where agent 0 is forced to defect at step 50.

To verify that the TV penalty suppresses the punishment response we examine trained agents' behavior when one agent is forced to deviate. Figure 6 shows the resulting price trajectories for the no-intervention baseline and the TV-only condition at λ=0.5. Under no intervention, agent 1 follows agent 0 down, a coordinated drop characteristic of an SPC strategy, and both recover within a few steps. Under TV-only, agent 1 does not respond to the deviation, only agent 0 dips and rebounds alone. This indicates that CURB's TV penalty specifically eliminates the punishment response that sustains collusive SPC equilibria.

F.6 TV-distance training trajectories

We test whether CURB's reduction in collusion coincides with suppression of the punishment signal it targets. We train tabular Q-learning agents on Bertrand Calvano $n = 2$ under no intervention and under CURB with $\lambda \in \{0.05, 0.1, 0.2, 0.5\}$, and log the TV statistic used in CURB's penalty throughout training. TV is averaged over each 10k-step logging window and over both agents.

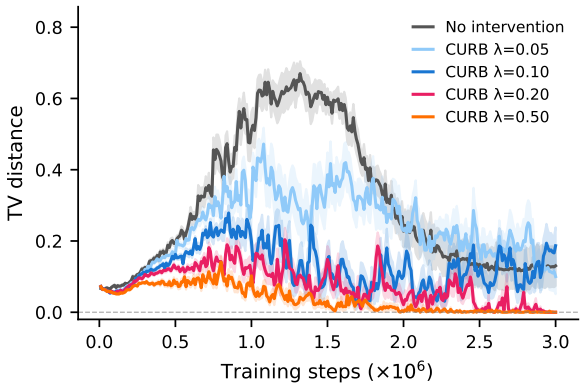


Figure 7: TV distance over training with varying λ.

Figure 7 plots TV over training. Under no intervention, TV rises sharply as agents learn reward-punishment strategies, peaking at 0.67 around 1.3×10^6 steps. CURB suppresses this rise monotonically in λ: peak TV falls to 0.47, 0.28, 0.20 and 0.14 for $\lambda = 0.05, 0.1, 0.2, 0.5$. For $\lambda \geq 0.2$, TV is driven to essentially zero by the end of training. For $\lambda \leq 0.1$, the penalty dampens TV but does not eliminate it. This ordering matches the collusion results in our previous experiments. The λ values that drive TV to zero are those that reduce the Collusion Index to near-competitive levels, while the weaker penalties retain both residual TV and collusion. Baseline TV declines late in training. This reflects exploration decaying, so deviations rarely occur on-path within the TV window; it does not mean punishment has disappeared, and the forced-defection test (Appendix F.5) shows that baseline agents still punish deviations.

G Frequently Asked Questions

G.1 Can algorithmic collusion arise through mechanisms other than punishment threats, and does CURB address those mechanisms?

Prior work has identified mechanisms through which supra-competitive outcomes can arise without retaliation. One example is the sequential no-regret-learning mechanism of Arunachaleswaran et al. [2], in which a first-moving agent uses a no-regret algorithm and a second-moving agent optimizes against the resulting static environment. This can produce monopoly-like prices without a punishment threat. However, Hartline [20] contests whether this outcome constitutes collusion in the classical sense, since only the first mover behaves non-competitively while the second simply best-responds to the induced demand.

A second example is "spurious collusion" [14], in which insufficient exploration and optimistic initialization lead Q-learning agents to converge to supra-competitive prices without genuine retaliatory behavior. Calvano et al. [14] distinguish these outcomes from genuine collusion in the classical economic sense: high prices alone do not constitute collusion unless they are sustained by a reward–punishment scheme. They further show that optimistic-initialization-driven exploration can produce spurious outcomes when agents are memoryless or sufficiently myopic, whereas forward-looking agents that condition on past prices can learn genuinely collusive strategies under random exploration.

More generally, conditional invariance should not be equated with non-collusive behavior. It indicates only that an agent's action distribution does not vary depending on whether its opponent cooperates or defects. Under the assumptions of our theoretical results, zero conditional TV rules out the non-trivial punishment structure characterized by Proposition 2, but it does not establish that the resulting outcome is competitive or rule out supra-competitive outcomes sustained through other mechanisms.

CURB is deliberately designed for a narrower setting: simultaneous-move repeated games with history-encoding states, where collusive outcomes are sustained by retaliatory threats.

G.2 How can the defection criterion distinguish welfare-improving deviations from welfare-reducing ones?

Definition 3 restricts collusive profiles to those satisfying $W(Q_0) < W(Q^{Nash})$, thereby excluding welfare-optimal cooperation by definition. However, our experimental defection criterion does not itself distinguish welfare-reducing from welfare-optimal cooperative paths. Definition 4 leaves ϕ_i general, allowing a designer to target only deviations from welfare-reducing paths, although our current experiments do not exploit this capability.

In market settings, one possible criterion uses consumer surplus directly: $\phi_i(s, a_i) = \mathbf{1}[CS(s, a_i) > \overline{CS}]$, where CS is computed from a known or estimated demand function and $\overline{CS}$ is a reference level, such as a rolling historical average. This classifies actions associated with increased consumer surplus as departures from collusive behavior. Conversely, if agents sustain a welfare-optimal cooperative path, a unilateral deviation that reduces consumer surplus remains unflagged. Retaliation against such a deviation is therefore not treated as punishment for a collusive deviation and is not targeted by the TV penalty. Computing consumer surplus from demand models is standard in economics. Its use as a defection criterion within CURB is a proposed instantiation rather than an established approach in the algorithmic-collusion literature.

G.3 Does CURB address delayed or non-simple punishment strategies?

Our current detector measures the change in an agent's action distribution immediately following an opponent's deviation (lag-1). Thus, the current analysis and implementation focus on immediate-onset punishment and do not explicitly detect delayed punishment.

However, bounded rewards and a discount factor $\delta < 1$ imply that sufficiently delayed punishment cannot deter a profitable deviation. If a deviation yields a one-shot gain $\Delta > 0$, and per-period rewards lie in $[r_{\min}, r_{\max}]$, the maximum discounted payoff loss from punishment beginning (H) periods later is bounded by $\frac{\delta^H (r_{\max} - r_{\min})}{1-\delta}$. This bound converges to zero as (H) increases. Consequently, for any fixed game, a sufficiently large but finite horizon captures all punishment delays that could be relevant for deterring profitable deviations.

A straightforward extension of CURB would therefore compare conditional action distributions over the next (H) periods following a deviation, rather than only the immediately following action. This would accommodate delayed punishment within the relevant horizon.

G.4 How does CURB's belief injection relate to heterogeneous beliefs in Luo et al. [29]?

Luo et al. [29] model heterogeneous beliefs through initial Q-value scaling, where optimistic and pessimistic initializations reflect different priors about an opponent's likelihood to collude. CURB's belief injection can be viewed as a targeted, in-training analogue: rather than uniformly scaling the initial Q-table, it shifts Q-values for the specific contingency of whether a unilateral deviation goes unpunished, and is applied only to the agent being shaped.

H LLM Usage

Large language models (LLMs) were used during the development of this work. In particular, LLM-assisted code generation was used to help implement portions of the experimental infrastructure, including components of the simulation and evaluation pipeline. We used LLMs as a writing and editing aid during manuscript preparation, including assistance with prose drafting, LaTeX formatting, and proof writing. Technical content and ideation, experimental design, implementation and analysis, and scientific claims are the original work of the authors.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We formalize the connection between Calvano's empirical observation and Abreu's SPC theory. We propose CURB and empirically validate it across two repeated-game settings (Bertrand and Cournot competition) and two learning algorithms (tabular Q-learning and DQN).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We list all assumptions in the main part of the paper. For each theorem we provide a sketch and/or a full proof (in Appendix C).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose complete details for reproducibility in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is available at `https://github.com/ksivacha/Curb`

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We disclose complete details of our experiments in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All standard error across seeds are reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify compute details and time required for our runs in the Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We adhere to the NeurIPS Code of Ethics as detailed in the link.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We mention societal impacts of our work in the introduction(1), related work(2), and conclusion(7) sections.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit owners of any resource we use promptly.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code to run our experiments is well documented alongside the code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe and declare the scope of LLM usage in the Appendix H.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.