Learning Adversarially Robust Policies in Multi-Agent Games

Eric Zhao 1 2  Alexander R. Trott 2  Caiming Xiong 2  Stephan Zheng 2

Abstract

Robust decision-making in multiplayer games requires anticipating what reactions a player policy may elicit from other players. This is difficult in games with three or more players: when one player fixes a policy, it induces a subgame between the remaining players with many possible equilibria outcomes. Predicting the worst-case outcome of a policy is thus an equilibrium selection problem—one known to be generally NP-Hard. We show that worst-case coarse-correlated equilibria can be efficiently approximated in smooth games and propose a framework that uses the worst-case evaluation scheme to learn robust player policies. We further prove the framework can be extended to handle uncertainty about the bounded rationality of other players. In experiments, our framework learns robust policies in repeated N-player matrix games and, when applied to deep multi-agent reinforcement learning, can scale to complex spatiotemporal games. For example, it learns robust AI tax policies that improve welfare by up to 15%, even when taxpayers are boundedly rational.

1. Introduction

Robust decision-making in multi-agent settings requires performing well against the wide variety of behaviors that other strategic agents may adopt. In this paper, we study the problem of learning a robust policy for an agent (the “ego agent”) when other agents are at least boundedly rational. This learning task entails two important challenges (Figure 1). First, when other agents react to our ego agent’s policy choice, they may be choosing from many equally rational reactions. In fact, when there are at least two other agents, there may be infinitely many equilibria they can settle into in response. A robust policy should ensure our ego agent performs well in all such equilibria. Second, policies that do well against rational agents may not do well against boundedly rational agents; generalization is not guaranteed even if the agents are almost exactly alike (Pita et al., 2010). Therefore a robust policy must also ensure good performance against the equilibria of boundedly rational agents.

We propose a robust decision-making framework that overcomes these challenges by maximizing worst-case performance over all approximate coarse-correlated equilibria (CCE). Although approximating worst-case CCE is known to be generally hard (Papadimitriou & Roughgarden, 2008; Barman & Ligett, 2015), we prove that we can efficiently approximate worst-case CCE in smooth games. We further prove that by ensuring robustness to approximate CCE, our framework learns strategies that generalize to boundedly rational agents with differing incentives and reputation.

Our motivating application is mechanism design (“reverse game theory”), where an agent implements the rewards and dynamics (the “mechanism”) that other agents optimize for (Myerson, 2016). Traditional mechanism design has been limited to problems with a convenient mathematical structure, e.g., simple auctions, where the potential equilibria behavior of agents are known in closed-form. Recent computational and ML research has designed mechanisms for more general problems by predicting equilibria behaviors through agent-based modeling (Holland & Miller, 1991;
Bonabeau, 2002; Dütting et al., 2019) and multi-agent reinforcement learning (Zheng et al., 2020). By approaching mechanism design as a multi-follower Stackelberg game, our work establishes a theoretically sound foundation for learning robust mechanisms for general problems.

Our novel contributions include the following.

1. We prove the existence of polynomial time algorithms for approximating worst-case coarse-correlated-equilibria (CCE) in smooth games. This weakens prior hardness results by (Papadimitriou & Roughgarden, 2008; Barman & Ligett, 2015) which states such approximations are generally NP-Hard. We thus show that robust optimization over CCEs can be tractable.

2. We derive a robust strategic learning objective that can be viewed as solving a CCE-based multi-agent extension of robust Stackelberg equilibria. We prove it affords robustness to the different equilibria behaviors that may arise from agents of various incentives, bounded rationality, and reputation.

3. We propose a theoretically grounded uncoupled framework for learning robust strategies in games, and demonstrate it learns robust dynamic mechanisms when instantiated with deep multi-agent RL algorithms. In spatiotemporal economic simulations used by the AI Economist (Zheng et al., 2020), we show that robust dynamic tax policies can improve welfare by up to 15% and generalize to a range of boundedly rational agents.

2. Related Work

Finding Equilibria. A central goal of multi-agent learning is finding equilibria (or more generally solution concepts) which are sets of agent behavioral policies that are game-theoretically optimal (according to the definition of equilibrium). Prior work has used gradient-based methods, e.g., deep reinforcement learning, to find (approximate) equilibria with great success in multiplayer games such as Diplomacy (Gray et al., 2021) and in training Generative Adversarial Networks (Schäfer et al., 2020). However, learning robust strategies in multi-agent games is an open problem (Pita et al., 2010; Kiekintveld et al., 2013). Similarly, Bayesian Stackelberg Games assume a prior over a space of possible follower utility functions (Conitzer & Sandholm, 2006). We extend on this by considering a general setting with multiple, possibly interacting, followers.

3. Problem Formulation

Notation. We consider a general-sum game $G$ with $N + 1$ agents. The ego agent is index $i = 0$ and the other agents are $i = 1, \ldots, N$. $A_i$ denotes the set of $m_i$ actions for agent $i$, and $m = \sum_{i=1}^{N} m_i$. $\prod(A_i)$ is the set of probability distributions over action set $A_i$. The joint action set is $A_{1:n} := \prod_{i \in [1, \ldots, N]} A_i$, $P(A_{1:n})$ is the set of joint probability distributions over $A_{1:n}$, and $\prod_{A_{1:n}}$ is the set of product probability distributions over the action set $A_{1:n}$.

Bold quantities denote vectors of size $n$, with each component corresponding to an agent $i = 1, \ldots, N$. For example, $a \in A_{1:n}$ denotes an action vector over all agents except agent 0 while $a_{-i}$ denotes the profile of actions chosen by all agents except agents 0 and $i$.

Every agent $i = 0, \ldots, n$ may have a utility function $u_i : A_0 \times A_{1:n} \rightarrow \mathbb{R}$ with bounded payoffs. For example, $u_i(a_0, a)$ denotes the utility of agent $i$ under action $a_0 \in A_0$ by agent 0 and actions $a \in A$ by agents 1, . . . , $n$. When $a_0$ is clear from context, we’ll write $u_i(a)$, suppressing $a_0$. We denote expected utility as $\pi_i(x_0, x) := \mathbb{E}_{a_0 \sim x_0, a \sim x} u_i(a_0, a)$ where again we write $\pi_i(x)$ when $x_0$ is clear from context.

Succinct Games. To derive our complexity results, we will use standard assumptions on our game $G$ so that working with equilibria is not trivially hard (Papadimitriou & Roughgarden, 2008). We assume $G := (I, T, U)$ is a succinct game, i.e., has a polynomial-size string representa-
tion. Here, \(I\) are efficiently recognizable inputs, and \(T\) and \(U\) are polynomial algorithms. \(T\) returns \(n, m_0, \ldots, m_N\) given inputs \(z \in I\), while \(U\) specifies the utility function of each agent. We assume \(G\) is polynomial type, i.e., \(n, m_0, \ldots, m_N\) are polynomial bounded in \(|I|\). Also, we assume that \(G\) satisfies the polynomial expectation property, i.e., utilities \(\pi(x_0, x)\) can be computed in polynomial time for product distributions \(x\). The latter assumption is known to hold for virtually all succinct games of polynomial type (Papadimitriou & Roughgarden, 2008). These assumptions are necessary to ensure that simply evaluating the payoff of a coarse-correlated equilibria does not require superpolynomial time. All complexity results in our work, including Theorem 1, and cited results from prior works are under these assumptions. Later, we will use additional “smoothness” conditions on \(G\) that allow us to overcome prior hardness results about succinct games.

**Objective.** We aim to learn a policy \(x_0 \in P(A_i)\) that maximizes our ego agent’s expected utility \(\pi_0(x_0, x)\), where \(x\) denotes the policies played by other agents. At test time, the policy \(x \in P(A_1, \ldots, A_N)\) may be decided in response to our selection of \(x_0\). For example, the ego agent \(i = 0\) may be a policymaker setting a tax policy \(x_0\) that maximize social welfare as measured by \(u_0\), e.g., total utility \(u_0 = \sum_{i=1}^{N} u_i\). In response, the tax-payers play \(x\), e.g., they choose how much to work and income to report, given tax policy \(x_0\).

A robust policy maximizes \(u_0\) even if the other agents follow a policy \(x\) that was previously unseen or not anticipated. For instance, an ego agent may have trained assuming the other agents are rational, but faces boundedly rational agents at test time. To define robustness in this setting, we use uncertainty sets \(X\) over the joint distributions \(P(A)\), where \(X(x_0) \subseteq P(A)\) denotes a set of potential behaviors of other agents in response to \(x_0\). Here, “uncertainty” refers to the need to anticipate a range of behaviors that could be expected at test time. Given an “appropriate” uncertainty set \(X^{\text{train}}(x_0)\), the abstract robustness objective is then to solve:

\[
\max_{x_0} \min_{x \in X^{\text{test}}(x_0)} \pi_0(x_0, x). \tag{1}
\]

Here, “appropriate” refers to uncertainty sets that yield good generalization. In other words, when used to learn a policy \(x_0\), an appropriate \(X^{\text{train}}\) maximizes the performance of \(x_0\) on a test set \(X^{\text{test}}\) of agent policies \(x\). Note that in general, the test set \(X^{\text{test}}\) may be different from the uncertainty set \(X^{\text{train}}\) used during training. For brevity, we will suppress the superscripts and \(X\) generally refers to \(X^{\text{train}}\).

This objective is challenging to solve: it features nested optimization, and it is non-trivial to characterize the behaviors of agents and “appropriate” uncertainty sets, especially when the game \(G\) has large state and action spaces. If the agent behaviors \(A\) are simple enough, one may use historical data or domain knowledge to construct uncertainty sets. In lieu of such priors, in this work, we derive uncertainty sets through game-theoretic reasoning about the rational behaviors a group of strategic agents may converge to.

### 4. Finding Uncertainty Sets

Our key design challenge is choosing an uncertainty set function \(X\). We have two desiderata:

**Tractability.** We want to efficiently sample (worst-case) behaviors from \(X\). That is, there exists an algorithm with runtime polynomial in the size of our game \(|I|\) to solve,

\[
\min_{x \in X(x_0)} \pi_0(x_0, x). \tag{2}
\]

**Generalization.** \(X\) is large enough such that a policy that solves the robustness objective (Equation 1) has good worst-case performance against any equilibria that may arise from boundedly rational agents.

We first review several candidate uncertainty sets \(X\) under the following assumptions (which we will later relax):

1. *A-priori*, we know the utility functions \(u_1, \ldots, u_N\).
2. Agents 1, \ldots, \(N\) are rational expected-utility maximizers that, as a group, will not “settle” into a choice of \(x\) where an agent can boost their expected utility by unilaterally changing their strategy.
3. We can credibly commit to a strategy \(x_0\), e.g. by being unable to react to the strategies \(x\) of other agents.

These assumptions justify choosing a uncertainty set based on the coarse-correlated equilibria (CCE) that our robust policy can induce between the other agents. We will prove that, at least in smooth games, approximate worst-case sampling of CCE can be done efficiently. We then extend the sampling scheme to ensure robustness to bounded rationality agents, by relaxing to an uncertainty set based on \(\epsilon\)-CCE; we prove that these approximate equilibria naturally yield generalization to a diverse variety of agents.

#### 4.1. Candidate Uncertainty Sets

**Dominant Strategies.** A natural choice of \(X\) is to extend Stackelberg equilibria to a multi-follower setting and define \(X(x_0)\) as the set of best responses to \(x_0\),

\[
X(x_0) = \{x \mid \forall i \in [1, \ldots, n], \tilde{x}_{-i} \in \tilde{P}(A)_{-i} : x_i \in \text{argmax}_{x_i} \pi_i(x_0, x_i, \tilde{x}_{-i})\}. \tag{3}
\]

However, this set is only non-empty when all agents have dominant strategies—a strong assumption that rarely holds when followers interact with one another, such as in mechanism design applications like auctions.
Learning Adversarially Robust Policies in Multi-Agent Games

Stability-Based Equilibria. Instead, we can define an uncertainty set over the stable outcomes that agents may converge to; this coincides with the uncertainty set in Equation 3 when it is non-empty.

\[
X(x_0) = \{x \in P(A) \mid \exists x_0 \in A_0 : (x_0, x) \in EQ\},
\]

where natural choices for EQ are mixed Nash equilibria (MNE) (which assume agents do not coordinate):

\[
\text{MNE} = \{x \in P^{\text{prod}}(A) \mid \forall i \in [1, \ldots, N], a_i \in A_i : E_{a \sim x}[u_i(a)] \geq E_{a \sim x}[u_i(\tilde{a}_i, a_{-i})]\}.
\]

or more general coarse-correlated equilibria (CCE):

\[
\text{CCE} = \{x \in P(A) \mid \forall i \in [1, \ldots, N], \tilde{a}_i \in A_i : E_{a \sim x}[u_i(x)] \geq E_{a \sim x}[u_i(\tilde{a}_i, a_{-i})]\}.
\]

Here, coarse-correlated equilibria describe more general joint strategies for the agents \(i = 1, \ldots, N\), such as coordination based on shared information.

Computational Hardness. Unfortunately, optimizing the robustness objective in Equation 1 is neither tractable with MNE nor CCE. Finding a worst-case MNE/CCE for a utility function \(u_0\) is equivalent to the NP-Hard problem of finding a MNE/CCE that maximizes a linear objective \(\nu\); simply finding any CCE that does not minimize \(\nu\) is NP-hard (Barman & Ligett, 2015). Specifically, consider the decision problem \(\Gamma\) of determining whether a game \(G\) (under our assumptions) admits a coarse correlated equilibrium \(x\) such that,

\[
\pi(x) > \min_{x \in \text{CCE}} \pi(x).
\]

This problem is NP hard for some choices of \(\nu\), including the social welfare function (Barman & Ligett, 2015). For our purposes, this implies that even sampling an approximately worst-case equilibria is intractable. This means that it is generally impossible to efficiently evaluate our ego agent’s policy as it is intractable to guarantee sampling anything other than uninformative equilibria behavior.

4.2 Smooth Games and Tractable Uncertainty Sets

We now prove that such hardness results do not always apply in smooth games. Specifically, we can sample a CCE that approximately maximizes \(\nu\)—or similarly, minimizes \(u_0\)—with run-time polynomial in the game size and smoothness. The first key step is reducing the hardness of finding a \(\nu\)-maximizing CCE to that of finding a Blackwell approachability halfspace oracle (Blackwell, 1956). Similar reductions have been described by (Jiang & Leyton-Brown, 2011) and (Barman & Ligett, 2015).

Lemma 1. There exists an algorithm that, given \(y \leq \max_{x \in \text{CCE}} \pi(x)\), returns an \(\epsilon\)-CCE \(x\) with \(\pi(x) \geq y - \epsilon\), with time complexity \(\text{Poly}(1/\epsilon, n, m)\) assuming a halfspace oracle that, given a vector \(\beta \in \mathbb{R}^{1+m}_+\) with non-negative components, returns an \(x \in P(A)\) such that \(\beta v(x) \leq 0\) in \(\text{Poly}(1/\epsilon, n, m)\), where,

\[
v(x) = \begin{bmatrix}
\pi_1(1, x_{-1}) - \pi_1(x) \\
\vdots \\
\pi_N(1, x_{-N}) - \pi_N(x) \\
\pi_N(m_N, x_{-N}) - \pi_N(x) \\
y - \pi(x)
\end{bmatrix}
\]

Unfortunately, the NP-hardness of finding non-trivial CCEs implies no such polynomial-time halfspace oracle exists for any \(y > \min_{x \in \text{CCE}} \pi(x)\). We will exploit the structure, specifically smoothness (Roughgarden, 2015), of our game \(G\) to overcome this hardness.

No-Regret Dynamics as a Half-Space Oracle. We construct a cost-minimization game \(\tilde{G}\) by modifying our original game \(G\) such that each agent’s utility function \(u_i\) is replaced with \(-\nu\). In the proof of the following theorem, we show that the halfspace oracle’s optimization task can be reduced to optimizing social welfare in a game with a robust price of anarchy (POA, Definition A.1) of \(\rho := \max\{\rho_G, \rho_G\}\), where \(\rho_G\) is the robust POA of \(G\) and \(\rho_G\) is that of the constructed game \(\tilde{G}\). Thus, we can use no-regret dynamics (Cesa-Bianchi & Lugosi, 2006; Foster & Vohra, 1997; Hart & Mas-Colell, 2000) in this social welfare game to approximate the half-space oracle.

Lemma 2. Let \(\rho\) denote an upper bound on the robust price of anarchy of \(G, \tilde{G}\). There exists a Poly(1/\(\epsilon\), n, m, \(\rho\))-time halfspace oracle that, given a vector \(\beta \in \mathbb{R}^{1+m}_+\) and \(y \leq \rho \max_{x \in \text{CCE}} \pi(x)\), returns an \(x \in P(A)\) such that \(\beta v(x) \leq \epsilon\) where \(v\) is defined as in Equation 8.

By nesting the no-regret dynamics of Lemmas 1 and 2, we obtain our main result.

Theorem 1. For succinct \(n\)-agent \(m\)-action games of polynomial type and expectation property, there exists a Poly(1/\(\epsilon\), n, m, \(\rho\))-time algorithm that will find an \(\epsilon\)-CCE \(x\) with

\[
\pi(x) \geq \frac{1}{\rho} \max_{x \in \text{CCE}} \pi(x) - \epsilon,
\]

where \(\rho\) is the robust price of anarchy.
Informally, this result states that it is tractable to find a CCE that maximizes \( \nu \) up to the robust price of anarchy. While this relationship is immediate when \( \nu \) is the social welfare function (Roughgarden, 2015), our result shows that we can prove a similar relationship concerning the optimization of CCE against any linear function. This positive result allows for translation between well-known robust price-of-anarchy bounds and bounds on the tractability of CCE optimization.

**Corollary 1.** In weighted congestion games with linear cost functions (Bhawalkar et al., 2014), for any linear function \( \nu \), we can find an \( \epsilon \)-CCE \( x \) such that

\[
\rho(x) \geq 0.38 \max_{x \in \text{CCE}} \rho(x) - \epsilon,
\]

in \( \text{Poly}(1/\epsilon, n, m) \) time.

While (Barman & Ligett, 2015) showed the decision problem \( \Gamma \) of finding a non-trivial CCE is NP-Hard in general, our Theorem 1 also shows it is tractable in games with a small robust price of anarchy.

**Corollary 2.** The decision-problem \( \Gamma \) is in P for games where \( \frac{1}{\rho} \max_{x \in \text{CCE}} \rho(x) > \min_{x \in \text{CCE}} \rho(x) \).

Motivated by these findings on the tractability of adversarial sampling from CCE and the remaining intractability of MNE, we propose to adopt an uncertainty set based on CCE. In the next section, we modify this choice of uncertainty set in the interests of generalizing to agents that violate our behavioral assumptions.

### 4.3. Weakening Assumptions and Generalization

We now consider weaker variants of our key assumptions:

1. **Subjective rationality:** At test time, an agent’s utility function \( u_i \) may differ from the anticipated utility function \( \bar{u}_i \) (Simon, 1976). Many models of subjective rationality, such as Subjective Utility Quantal Response (Nguyen et al., 2013), bound the difference between \( u_i \) and \( \bar{u}_i \) as \( ||u_i - \bar{u}_i||_\infty \leq \gamma_s \) with \( \gamma_s > 0 \).

2. **Procedural rationality:** An agent may try to but not fully succeed in maximizing their utility (Simon, 1976). This property means an agent could gain up to \( \gamma_p > 0 \) utility if they unilaterally deviate, for example.

3. **Myopia:** An agent may possess commitment power or otherwise be non-myopic, factoring in long-term incentives with a non-zero discount factor \( \gamma_m \in (0, 1) \). This relates to notions of exogenous commitment power, e.g., partial reputation, in Stackelberg games (Kreps & Wilson, 1982; Fudenberg & Levine, 1989).

These variations represent common forms of bounded rationality. We now show that the sampling scheme we devise for Theorem 1 can be extended to maintain robustness despite these weaker assumptions. We aim to learn strategies \( x_0 \) that perform well in the presence of agents with these variations. Hence, we aim to use uncertainty sets \( X \) that encode such behaviors. The next proposition suggests it suffices to simply relax our uncertainty set \( X \) to include more approximate equilibria.

**Proposition 1.** The uncertainty set \( X' \) of (any combination of) agents violating assumptions 1-3 with parameters \( \gamma_m, \gamma_s, \gamma_p \) is contained in the set of \( \epsilon \)-CCE:

\[
\text{CCE}_\epsilon = \{ x \in P(A) \mid \forall i \in [1, \ldots, N], \bar{a}_i \in A_i : \epsilon_i \geq \max_{a \sim \text{CCE}} [u_i(a) + \epsilon_i] \},
\]

where \( \epsilon_i = \max \{ ||u_i||_\infty, \gamma_s, 2\gamma_p \} \). Hence, we can train policies robust to such agents by using:

\[
X_\epsilon(x_0) = \{ x \in P(A) \mid \exists x_0 \in A_0 : (x_0, x) \in \text{CCE}_\epsilon \}
\]

in the robustness objective of Equation 1.

### 4.4. Finding Worst-Case \( \epsilon \)-CCE

We synthesize these insights into Algorithm 3, which samples approximately optimal \( \epsilon \)-CCE with polynomial runtime. By relaxation of Lemma 1, it yields the same optimality and runtime guarantees as Theorem 1, but over the set \( \epsilon \)-CCE rather than the set CCE. Some key characteristics are:

1. It only requires bandit feedback access to utility functions, which is efficient under the polynomial expectation property of \( \pi_0, \ldots, \pi_N \).

2. It is uncoupled: an agent only needs access to their own utility function and none of the other agent’s.

3. Sampling worst-case members of \( \epsilon \)-CCEs is not significantly harder than sampling worst-case members of strict CCEs (as guaranteed by Theorem 1).

In sum, Algorithm 3 (with \( \nu = -u_0 \)) is a theoretically sound method for sampling pessimistic outcomes from our uncertainty set \( X \). Its key characteristics will allow us to build on Algorithm 3 in the next section to devise scalable variants for complex dynamic mechanism design problems. In particular, Algorithm 3’s dependence on only bandit feedback allows us to substitute in blackbox learning algorithms, such as deep reinforcement learning. In addition, the uncoupled nature of the algorithm allows distributed implementation.

### 5. A Blackbox Robustness Framework

We now propose a scalable framework around our theoretically sound procedure for pessimistic sampling from \( X \). We
first describe two simplifications for the less practical components of Algorithm 3, introducing Lagrange multipliers and using scalable black-box-compatible sub-procedures. We then validate our modified Algorithm 3 in complex sequential decision-making games.

5.1. Simplifying Algorithm 3

Removing binary search. The binary search over \(y\) is a theoretically efficient search over possible optimal or worst-case values of \(\varepsilon\)-CCE. However, it is inefficient in a nested optimization like Equation 1. Observing that \(y\)’s value affects only the parameterization of the importance weight \(\beta_{m+1}\), we can fix \(v_{m+1}\) to a sufficiently small value such that \(\beta_{m+1} = 1[v_1, \ldots, v_m \leq 0]\).

Streamline Blackwell’s algorithm. We can merge components of \(v\) corresponding to the same agent, yielding a functionally equivalent alternative to Eq 8,

\[
v(x) = \begin{bmatrix}
\max_{a_1 \in A_1} \pi_1(a_1, x_{-1}) - \pi_1(x) \\
\vdots \\
\max_{a_n \in A_n} \pi_n(a_n, x_{-n}) - \pi_n(x)
\end{bmatrix}_{\sigma \to 0}
\]

(13)

This is more tractable than Eq 8 when \(A\) is, e.g., combinatorially large because we can efficiently approximate Eq 13. Specifically, we can approximate the regret components with local methods rather than explicitly enumerating all possible action deviations.

These two simplifications reveal that our algorithm (Alg 3) is equivalent to computing an upper-bound for the Lagrangian dual problem, \(L(\varepsilon)\), of adversarial sampling (Eq 18),

\[
L(\varepsilon) = \min_{x \in P(A)} \max_{\lambda} \pi_0(x) - \sum_{i=1}^{N} \lambda_i \left[\text{Reg}_i\left(x\right) - \varepsilon\right],
\]

(14)

\[
\text{Reg}_i\left(x\right) := \max_{a_i \in A_i} \pi_i(a_i, x_{-i}) - \pi_i(x),
\]

specifically, using decoupled no-regret dynamics to efficiently approximate the outer optimization \(\min_{x \in P(A)}\). In this sense, our theoretical results can be interpreted as a formal bound on how much decoupled approximations of \(\min_{x \in P(A)}\) affect the lower-bound that this dual problem gives us. Specifically, Theorem 1 suggests that when playing a sufficiently smooth game it is reasonable to use decoupled algorithms to approximate the dual problem. In Algorithm 1, we describe a general framework for sampling worst-case CCE that builds on this key takeaway. This framework generalizes the role of no-regret algorithms in our decoupled optimization to support plugging in any blackbox algorithm.

Algorithm 1 Decoupled sampling of pessimistic equilibria.

**Output:** Approximate lower-bound on \(L(\varepsilon)\) (Eq 14).

**Input:** Number of training steps \(M_s\) and self-play steps \(M_a\), reward slack \(\varepsilon\), multiplier learning rate \(\alpha\).

**Input:** Uncoupled self-play algorithm \(B\), regret estimators \(R_i: P(A) \rightarrow \mathbb{R}\) for each player \(i = 1, \ldots, N\).

Initialize mixed strategy \(x_1\).

**for** \(j = 1, \ldots, M_s\) \n
**for** \(i = 1, \ldots, N\) \n
Estimate regret \(r_i\) for agent \(i\): \(\hat{r}_i \leftarrow R_i(x_j)\), where \(r_i := \max_{x_i \in P(A_i)} \pi_i(\hat{x}_i, x_{-i}) - \pi_i(x)\).

Compute multiplier \(\lambda_i \leftarrow \lambda_i - \alpha \lambda (\hat{r}_i - \varepsilon)\).

**end for**

Use \(B\) to run \(M_s\) rounds of \(N\)-agent self-play with utilities \(\hat{u}_i(a) := (\lambda_i u_i(a) - u_0(a))/(1 + \lambda_i)\).

Set \(x_{j+1}\) as the resulting empirical play distribution.

**end for**

Return \(\frac{1}{M_s} \sum_{t=1}^{M_s} \pi_0(x_t)\).

5.2. Empirical Results on Finding \(\varepsilon\)-CCE

Motivated by dynamic mechanism design, we now use RL as a black-box sub-procedure and show empirically that this enables finding adversarial CCEs in challenging sequential games. We instantiate Algorithm 1 with multi-agent proximal policy optimization (Schulman et al., 2017) for \(B\), and a Monte-Carlo regret estimator to obtain Algorithm 4.

Sequential Bimatrix Games. We analyze the behaviors that Algorithm 4 samples in an extension of the classic repeated bimatrix game (Figure 2), whose Nash equilibria can be solved efficiently and is well-studied in game theory. At each timestep \(t\), a row (agent 1) and column player (agent 2) choose how to move around a \(4 \times 4\) grid, while receiving rewards \(r_1(s_i, s_j), r_2(s_i, s_j)\). The current location is at row \(s_i\) and column \(s_j\). The row (column) player chooses whether to move up (left) or down (right). We select the payoff matrices \(r_1\) and \(r_2\), illustrated in Figure 2, so that only one Nash equilibrium exists and that the equilibrium constitutes a “tragedy-of-the-commons,” where agents selfishly optimizing their own reward leads to less reward overall. We also introduce a passive gambler that observes the game and receives a payoff \(r_0(s_i, s_j)\). The gambler does not take any actions and its payoff is constructed such that its reward is high when the agents are at the Nash equilibrium. For the sake of validating Algorithm 4, we treat the gambler as the ego agent, but note that for this analysis we are only interested in the behaviors of agents 1 and 2.

In effect, this toy setting allows us to verify that our algorithm samples realistic worst-case behaviors—that is, that agents 1 and 2 learn to deviate from their tragedy-of-the-commons equilibrium in order to reduce the gambler’s re-
ward but also without significantly increasing their own regret. For small values of \( \epsilon \), our algorithm also discovers the Nash equilibrium. Because \( \epsilon \) acts as a constraint on agent regret, larger values of \( \epsilon \) enable our algorithm to deviate farther from the Nash equilibrium, discovering \( \epsilon \)-equilibria to the bottom-right that result in lower gambler rewards.

Deviations from a Nash equilibrium should yield higher regret, i.e., regret should increase with \( \epsilon \). Figure 2 (right) clearly shows this trend under our algorithm. As described by the robustness objective in Equation 1, sampling these adversarial agent behaviors is key to training a ego agent.

6. Optimizing Adversarial Robustness

We now use our worst-case equilibria sampling algorithm (Algorithm 4) in the inner loop of our robustness objective (Equation 1). We evaluate the full method on 1) repeated matrix games and 2) a dynamic mechanism design task. In each case, we evaluate the robustness of \( x_0 \) on agents with modified utility functions, e.g., with risk aversion.

6.1. Repeated Matrix Games

We extend the repeated bimatrix game with a gambler which actively participates as another player. The setting is now a 4-player, general-sum, normal-form game on a randomly generated \( 7 \times 7 \times 7 \times 7 \) payoff matrix, with the same action and payoff rules as shown in Figure 2. In this game, we consider the following (non-gambler) agent behaviors:

1. Behaviors from \textit{vanilla RL agents}. These agents experience the original reward function \( r_i \).

2. Behaviors from \textit{adversarial RL agents}. Adversarial agents (Adv) experience a modified reward \( r_i' = r_i - Q r_0 \) where \( Q \) parameterizes how adversarial they are: larger \( Q \) equates greater adversarialness.

3. Behaviors from \textit{risk-averse agents}. Risk-averse agents (RiskAv) experience a modified reward \( r'_i = r_i^{-\eta} - 1 - \eta \), where \( \eta > 0 \) parameterizes their risk aversion with higher values equating to greater risk aversion.

4. Behaviors sampled by our algorithm, Algorithm 4.

Figure 3 shows the average rewards of gamblers trained (rows) and evaluated (columns) on each of these types of agents. We make three key observations. First, gamblers trained on one type of agent generally perform better when evaluated on the same type of agent (diagonal entries). Second, gamblers trained with our adversarial sampling scheme perform better across the board. Third, the robustness gains of our adversarial sampling scheme are stronger when \( \epsilon \) is large. This is expected as \( \epsilon \) parameterizes the adversarial strength of our sampling scheme. For small \( \epsilon \), adversarial sampling reduces to random sampling as the CCE constraint is so tight it permits no adversarial deviations. We also observed that, even though all methods were run with 20 seeds and filtered down to 10 seeds on a validation set, our algorithm results remain somewhat noisy. This is because our algorithm can fail to converge when badly initialized.
6.2. Dynamic Mechanism Design

We now evaluate our algorithm in a dynamic mechanism design problem: tax policy design with interacting strategic agents in a spatiotemporal economic simulation (Zheng et al., 2020). See Appendix B.1 for a visualization. In this simulation, the ego agent sets the tax policy, and the other agents play a partially observable game, given the tax policy. Tax-payers earn income $z_{i,t}$ from labor $l_{i,t}$ and pay taxes $T(z_{i,t})$. They optimize their expected utility $\mathbb{E}_{x_{0:t}}[\sum_{t=0}^T u_{i,t}]$, using isoelastic utility (Arrow, 1971):

$$\tilde{x}_{i,t} = z_{i,t} - T(z_{i,t}), \quad \tilde{z}_{i,t} = \sum_{t' \leq t} \tilde{x}_{i,t'}, \quad (15)$$

$$r_{i,t}(\tilde{x}_{i,t}, l_{i,t}) = \frac{x_{i,t}^\eta - 1}{1 - \eta} - l_{i,t}, \quad \eta > 0, \quad (16)$$

where $\tilde{x}_{i,t}$ is the post-tax endowment of agent $i$, and $\eta$ sets the degree of risk aversion (higher $\eta$ means higher risk aversion). The ego agent optimizes social welfare $swf$:

$$swf := 1 - \frac{N}{N-1} \cdot gini(z) \cdot \prod_{i=1}^N \frac{z_i}{\epsilon_i(z)} \cdot \frac{\sum_{i=1}^N z_i}{\prod_{i=1}^N \epsilon_i(z)} \cdot \frac{z_i}{\epsilon_i(z)} \cdot \frac{\sum_{i=1}^N z_i}{\prod_{i=1}^N \epsilon_i(z)} \cdot \frac{z_i}{\epsilon_i(z)}$$

(17)

Figure 4 shows the social welfare achieved by our algorithm, AI Economist policies (i.e., naive multi-agent RL) (Zheng et al., 2020), and baseline tax schemes (Saez, US Federal). Naive multi-agent RL policies achieve good test performance when evaluated on the same agents seen in training, but perform poorly against agents of different risk-aversion and noise level. They are often outperformed by baseline tax heuristics, which perform surprisingly well under strong risk aversion ($\eta = 0.27$) and noisy agents (entropy bonus $\alpha = 0.25, 2.5$). This highlights that traditional tax models may be more robust than those found with machine learning, even in complex environments. However, our algorithm closes this robustness gap, consistently outperforming or tying both AI Economists and traditional tax models.

---

Figure 3. Robust performance in $N$-agent matrix games. We train an ego agent in a $7 \times 7 \times 7 \times 7$ matrix game with $n = 4$ agents (including the ego agent) until convergence. For each method, we train 20 seeds and select the top 10 in a validation environment. Each row corresponds to a specific agent type that the ego agent is trained on. ‘MARL’ refers to agents trained using their ‘Original’ reward definition; ‘Adv’ refers to adversarial agents; ‘RiskAv’ refers to risk-averse agents (see Section 6 for details). The ego agents trained on these types of agents tend to perform best when evaluated on the same type seen during training. In contrast, ego agents trained against agent behaviors sampled using our algorithm ($\epsilon = 50$) perform within standard error of top-1 on all agent types. We use the ‘Original’ reward definition when training with our algorithm.

| Training ↓ Testing → | Original | Adv ($Q = 0.25$) | Adv ($Q = 1$) | RiskAv ($\eta = 0.05$) | RiskAv ($\eta = 0.2$) |
|----------------------|----------|------------------|---------------|-----------------------|----------------------|
| Free Market          | 326±5    | 527±2            | 427±1         | 162±1                 | 248±2                |
| Federal              | 335±5    | 637±5            | 497±2         | 150±2                 | 270±1                |
| Saez                 | 381±1    | 597±3            | 487±4         | 189±4                 | 265±1                |
| Ours ($\epsilon = 30$) | 375±9    | 646±6            | 514±12        | 164±10                | 266±2                |
| AI Economist (Original) | 386±2    | 628±5            | 515±1         | 123±12                | 267±0                |
| AI Economist ($\eta = 0.11$) | 253±5    | 683±7            | 506±1         | 140±1                 | 255±2                |
| AI Economist ($\eta = 0.19$) | 308±17   | 665±9            | 543±6         | 82±29                 | 256±3                |
| AI Economist ($\eta = 0.27$) | 339±11   | 603±3            | 477±1         | 137±10                | 266±0                |
| AI Economist ($\alpha = 0.25$) | 324±2    | 625±10           | 501±7         | 121±25                | 263±0                |
| AI Economist ($\alpha = 2.5$) | 104±27   | 636±3            | 246±33        | 49±10                 | 251±4                |
7. Future Work

Our work emphasizes the efficient sampling of worst-case equilibria as a key bottleneck for robust decision-making methods in multi-agent games. Tight lower bounds for finding optimal CCE in smooth games are an important open problem in this topic. While our work explores game-theoretic selections of multi-agent uncertainty sets, future work may address uncertainty sets that leverage domain feedback and historical data.

References

Aghassi, M. and Bertsimas, D. Robust game theory. Mathematical Programming, 107(1):231–273, June 2006. ISSN 1436-4646. doi: 10.1007/s10107-005-0686-0. URL https://doi.org/10.1007/s10107-005-0686-0.

Arrow, K. J. The theory of risk aversion. Essays in the theory of risk-bearing, pp. 90–120, 1971. Publisher: Markham Chicago.

Barman, S. and Ligett, K. Finding Any Nontrivial Coarse Correlated Equilibrium Is Hard. ACM SIGecom Exchanges, 14, April 2015. doi: 10.1145/2845926.2845929.

Basilico, N., Coniglio, S., Gatti, N., and Marchesi, A. Bilevel programming methods for computing single-leader-multi-follower equilibria in normal-form and polymatrix games. EURO Journal on Computational Optimization, 8(1):3–31, March 2020. ISSN 2192-4414. doi: 10.1007/s13675-019-00114-8. URL https://doi.org/10.1007/s13675-019-00114-8.

Bhawalkar, K., Gairing, M., and Roughgarden, T. Weighted Congestion Games: The Price of Anarchy, Universal Worst-Case Examples, and Tightness. ACM Transactions on Economics and Computation, 2(4):14:1–14:23, October 2014. ISSN 2167-8375. doi: 10.1145/2629666. URL https://doi.org/10.1145/2629666.

Blackwell, D. An analog of the minimax theorem for vector payoffs. Pacific Journal of Mathematics, 6(1):1–8, January 1956. ISSN 0030-8730. URL https://msp.org/pjm/1956/6-1/pjm-v6-n1-p01-s.pdf. Publisher: Pacific Journal of Mathematics, A Non-profit Corporation.

Bonabeau, E. Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences, 99(suppl 3):7280–7287, May 2002. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.082080899. URL https://www.pnas.org/content/99/suppl_3/7280.

Cesa-Bianchi, N. and Lugosi, G. Prediction, Learning, and Games. January 2006. ISBN 978-0-521-84108-5. doi: 10.1017/CBO9780511546921. Journal Abbreviation: Prediction, Learning, and Games. Publication Title: Prediction, Learning, and Games.

Conitzer, V. and Sandholm, T. Computing the optimal strategy to commit. In Proceedings of the 7th ACM conference on Electronic commerce - EC '06, pp. 82–90, Ann Arbor, Michigan, USA, 2006. ACM Press. ISBN 978-1-59593-236-5. doi: 10.1145/1134707.1134717. URL http://portal.acm.org/citation.cfm?doid=1134707.1134717.

Daskalakis, C., Goldberg, P. W., and Papadimitriou, C. H. The Complexity of Computing a Nash Equilibrium. SIAM Journal on Computing, 39(1):195–259, January 2009. ISSN 0097-5397. doi: 10.1137/070699652. URL https://epubs.siam.org/doi/10.1137/070699652. Publisher: Society for Industrial and Applied Mathematics.

Dütting, P., Feng, Z., Narasimhan, H., Parkes, D., and Ravindranath, S. S. Optimal auctions through deep learning. In International Conference on Machine Learning, pp. 1706–1715, 2019.

Foster, D. P. and Vohra, R. V. Calibrated learning and correlated equilibrium. Games and Economic Behavior, 21(1):40–55, 1997. ISSN 0899-8256. doi: https://doi.org/10.1006/game.1997.0595. URL https://www.sciencedirect.com/science/article/pii/S0899825697905959.

Fudenberg, D. and Levine, D. K. Reputation and Equilibrium Selection in Games with a Patient Player. Econometrica, 57(4):759–778, 1989. ISSN 0012-9682. doi: 10.2307/1913771. URL https://www.jstor.org/stable/1913771. Publisher: [Wiley, Econometric Society].

Gini, C. Variabilità e mutabilità. 1912.

Gray, J., Lerer, A., Bakhtin, A., and Brown, N. Human-Level Performance in No-Press Diplomacy via Equilibrium Search. arXiv:2010.02923 [cs], May 2021. URL http://arxiv.org/abs/2010.02923. arXiv: 2010.02923.

Hart, S. and Mas-Colell, A. A simple adaptive procedure leading to correlated equilibrium. Econometrica, 68(5):1127–1150, 2000. doi: https://doi.org/10.1111/1468-0262.00153. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00153.
Learning Adversarily Robust Policies in Multi-Agent Games

Holland, J. H. and Miller, J. H. Artificial Adaptive Agents in Economic Theory. *American Economic Review*, 81 (2):365–71, 1991. URL https://econpapers.repec.org/article/aeaaecrev/v_3a81y3ay_3a1991_3ai_3a2_3ap_3a365-71.htm.

Hou, L., Pang, L., Hong, X., Lan, Y., Ma, Z., and Yin, D. Robust Reinforcement Learning with Wasserstein Constraint. arXiv:2006.00945 [cs, stat], June 2020. URL http://arxiv.org/abs/2006.00945.

Jiang, A. X. and Leyton-Brown, K. A General Framework for Computing Optimal Correlated Equilibria in Compact Games. In Chen, N., Elkind, E., and Koutsoupias, E. (eds.), *Internet and Network Economics*, Lecture Notes in Computer Science, pp. 218–229, Berlin, Heidelberg, 2011. Springer. ISBN 978-3-642-25510-6. doi: 10.1007/978-3-642-25510-6_19.

Kiekintveld, C., Islam, T., and Kreinovich, V. Security games with interval uncertainty. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, AAMAS ’13, pp. 231–238, Richland, SC, May 2013. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-1993-5.

Korzhyk, D., Conitzer, V., and Parr, R. Security games with multiple attacker resources. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume One*, IJCAI’11, pp. 273–279, Barcelona, Catalonia, Spain, July 2011. AAAI Press. ISBN 978-1-57735-513-7.

Kreps, D. M. and Wilson, R. Reputation and imperfect information. *Journal of Economic Theory*, 27(2):253–279, August 1982. ISSN 0022-0531. doi: 10.1016/0022-0531(82)90030-8. URL https://www.sciencedirect.com/science/article/pii/0022053182900308.

Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., and Russell, S. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4213–4220, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33014213. URL https://ojs.aaai.org/index.php/AAAI/article/view/4327. Number: 01.

Morimoto, J. and Doya, K. Robust Reinforcement Learning. In Leen, T. K., Dietterich, T. G., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems 13*, pp. 1061–1067. MIT Press, 2001. URL http://papers.nips.cc/paper/1841-robust-reinforcement-learning.pdf.

Myerson, R. B. Mechanism Design. In *The New Palgrave Dictionary of Economics*, pp. 1–13. Palgrave Macmillan UK, London, 2016. ISBN 978-1-349-95121-5. doi: 10.1057/978-1-349-95121-5_2675-1. URL https://doi.org/10.1057/978-1-349-95121-5_2675-1.

Nguyen, T. H., Yang, R., Azaria, A., Kraus, S., and Tambe, M. Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI’13, pp. 718–724, Bellevue, Washington, July 2013.

Nguyen, T. H., Sinha, A., Gholami, S., Plumpatre, A., Joppa, L., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., and Critchlow, R. CAPTURE: A new predictive anti-poaching tool for wildlife protection. *Proceedings of 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2016. URL https://ink.library.smu.edu.sg/sis_research/4660.

Papadimitriou, C. H. and Roughgarden, T. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14:1–14:29, August 2008. ISSN 0004-5411. doi: 10.1145/1379759.1379762. URL https://doi.org/10.1145/1379759.1379762.

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. Robust Adversarial Reinforcement Learning. arXiv:1703.02702 [cs], March 2017. URL http://arxiv.org/abs/1703.02702. arXiv: 1703.02702.

Pita, J., Jain, M., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., and Kraus, S. Using Game Theory for Los Angeles Airport Security. *AI Magazine*, 30:43–57, March 2009. doi: 10.1609/aimag.v30i1.2173.

Pita, J., Jain, M., Tambe, M., Ordóñez, F., and Kraus, S. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, October 2010. ISSN 0004-3702. doi: 10.1016/j.artint.2010.07.002. URL https://www.sciencedirect.com/science/article/pii/S0004370210000986.

Pita, J., John, R., Maheswaran, R., Tambe, M., and Kraus, S. A robust approach to addressing human adversaries in security games. In *ECAI 2012*, pp. 660–665. IOS Press, 2012.

Roughgarden, T. Intrinsic Robustness of the Price of Anarchy. *Journal of the ACM*, 62(5):32:1–32:42, November 2015. ISSN 0004-5411. doi: 10.1145/2806883. URL https://doi.org/10.1145/2806883.
Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. Trust Region Policy Optimization. arXiv:1502.05477 [cs], April 2017. URL http://arxiv.org/abs/1502.05477.

Schäfer, F., Zheng, H., and Anandkumar, A. Implicit competitive regularization in GANs. arXiv:1910.05852 [cs, stat], March 2020. URL http://arxiv.org/abs/1910.05852. arXiv: 1910.05852.

Simon, H. A. From substantive to procedural rationality. In Kastelein, T. J., Kuipers, S. K., Nijenhuis, W. A., and Wagenaar, G. R. (eds.), 25 Years of Economic Theory: Retrospect and prospect, pp. 65–86. Springer US, Boston, MA, 1976. ISBN 978-1-4613-4367-7. doi: 10.1007/978-1-4613-4367-7_6. URL https://doi.org/10.1007/978-1-4613-4367-7_6.

Tessler, C., Efroni, Y., and Mannor, S. Action Robust Reinforcement Learning and Applications in Continuous Control. arXiv:1901.09184 [cs, stat], May 2019. URL http://arxiv.org/abs/1901.09184.

Zheng, S., Trott, A., Srinivasa, S., Naik, N., Gruesbeck, M., Parkes, D. C., and Socher, R. The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies. arXiv:2004.13332 [cs, econ, q-fin, stat], April 2020. URL http://arxiv.org/abs/2004.13332.
A. Appendix: Proofs

A.1. Proof of Lemma 1

First, we observe that finding a $\nu$-maximizing CCE is equivalent to solving the linear program below. Writing the joint distribution $x$ as a $\prod_{i=1}^{N} m_i$-size vector of joint densities, a $\nu$-maximizing CCE is given by:

$$\max_{x \in P(A)} \mathcal{P}(x) \text{ s.t. } \forall i \in [1, \ldots, N], \forall a_i \in A_i :$$

$$\sum_{a' \in A} [u_i(a_i, a'_i) - u_i(a'_i)] x(a') \leq 0. \quad (18)$$

Although the number of constraints in this LP is polynomial, the number of variables is non-trivially exponential. In addition, the polynomial expectation property only guarantees polynomial time expected utility computations for product distributions.

Lemma 1. There exists an algorithm that, given $y \leq \max_{x \in \mathcal{P}(A)} \mathcal{P}(x)$, returns an $\epsilon$-CCE $x$ with $\mathcal{P}(x) \geq y - \epsilon$, with time complexity $\text{Poly}(1/\epsilon, n, m)$ assuming a halfspace oracle that, given a vector $\beta \in \mathbb{R}^{m+1}$ with non-negative components, returns an $x \in P(A)$ such that $\beta x \leq 0$ in $\text{Poly}(1/\epsilon, n, m)$, where,

$$v(x) = \begin{bmatrix} \pi_1(1, x_{-1}) - \pi_1(x) \\ \vdots \\ \pi_1(m_1, x_{-1}) - \pi_1(x) \\ \vdots \\ \pi_N(1, x_{-N}) - \pi_N(x) \\ \vdots \\ \pi_N(m_N, x_{-N}) - \pi_N(x) \\ y - \mathcal{P}(x) \end{bmatrix} \quad (8)$$

Proof. We will prove our lemma constructively. This proof closely mirrors (Cesa-Bianchi & Lugosi, 2006)’s proof of Blackwell approachability.

First, we establish approachability conditions. Note that $x$ is an $\epsilon$-CCE satisfying $\mathcal{P}(x) \leq \epsilon + y$ if $x$ satisfies $\nu(x) \leq \epsilon$ component-wise; the first $m$ components give that $x$ is an $\epsilon$-CCE and the last component gives that $\mathcal{P}(x) \leq \epsilon + y$. Thus if $y \leq \max_{x \in \mathcal{P}(A)} \mathcal{P}(x)$, then we know there exists a $x \in P(A)$ such that $\nu(x) \leq 0$. In other words, the closed convex set $\nu(x) \leq 0$ is approachable.

For the remainder of this proof, we assume $y \leq \max_{x \in \mathcal{P}(A)} \mathcal{P}(x)$ and prove that we will return an $\epsilon$-CCE that satisfies $\mathcal{P}(x) \leq \epsilon + y$. If $y$ does not satisfy this condition, then we will not return a CCE $x$ that satisfies $\mathcal{P}(x) < y$ because no such $x$ would exist. Let $x^* := \arg\max_{x \in \mathcal{P}(A)} \mathcal{P}(x)$.

Fix a choice of $\epsilon$. Now, consider the following iterative algorithm where we index timesteps by $t = 1, 2, \ldots$ and produce a sequence of mixed strategies beginning with some arbitrary choice of $x^{(0)} \in P(A)$. We now consider every timestep $t$ where the average value of $\nu(x)$,

$$\bar{v}_t := \frac{1}{t-1} \sum_{i=1}^{t-1} \nu(x^{(i)}), \quad (19)$$

does not satisfy $\bar{v}_t \leq \epsilon$ component-wise. If instead $\bar{v}_t \leq \epsilon$ holds component-wise, we can return $x^*$, a uniform distribution over $x^{(1)}, \ldots, x^{(t-1)}$, and be done.

Otherwise, project $\bar{v}_t$ onto the negative orthant, $S := [-\infty, 0]^{1+m}$, to obtain a vector $\beta \in \mathbb{R}^{m+1}$,

$$a_t = \arg\min_{a \in \mathbb{R}^{m+1}} \|\bar{v}_t - a\| \quad (20)$$

$$\beta_t = \frac{\bar{v}_t - a_t}{\|\bar{v}_t - a_t\|}. \quad (21)$$

Since $S$ is closed and convex, $\beta$ exists and is unique. Note that $\beta$ must be non-negative in all components by definition of $S$. We then consult our halfspace oracle to return our next iterate $x^{(t)} \in P(A)$ where $\beta_t x^{(t)} \leq 0$. A solution must exist as, for any $t$, $x^*$ satisfies $\beta_t x^{(t)} \leq 0$.

We now prove this algorithm returns the desired output in $\text{Poly}(\frac{1}{\epsilon}, n, m)$ and hence $\text{Poly}(\frac{1}{\epsilon}, |I|)$ time.

Since $\bar{v}_{t+1} = \frac{t-1}{t} \bar{v}_t + \frac{1}{t} \nu(x^{(t)})$, we may write,

$$d(\bar{v}_{t+1}, S)^2 = \|\bar{v}_{t+1} - a_{t+1}\|^2 \leq \|\bar{v}_{t+1} - a_t\|^2$$

$$= \|\frac{t-1}{t} \bar{v}_t + \frac{1}{t} \nu(x^{(t)}) - a_t\|^2$$

$$= \left(\frac{t-1}{t}\right)^2 \|\bar{v}_t - a_t\|^2$$

$$+ \left(\frac{1}{t}\right)^2 \|\nu(x^{(t)}) - a_t\|^2$$

$$+ 2 \frac{t-1}{t^2} \nu(x^{(t)}) - a_t \cdot (\bar{v}_t - a_t) \quad (23)$$

Let $u_{\text{max}} := \max\{\|u_1\|_{\infty}, \ldots, \|u_N\|_{\infty}, \|\nu\|_{\infty}\}$ and $u_{\text{min}} := -\max\{-\|u_1\|_{\infty}, \ldots, -\|u_N\|_{\infty}, -\|\nu\|_{\infty}\}$. Recall that by construction of $G$, $u_{\text{max}}, u_{\text{min}}$ is polynomial bounded by $|I|$ so without loss of generality, let us fix utilities to a unit ball: $u_{\text{max}} \leq 1, u_{\text{min}} \geq -1$. We can rearrange the equation to obtain,

$$l^2 \|\bar{v}_{t+1} - a_{t+1}\|^2 - (t-1)^2 \|\bar{v}_t - a_t\|^2$$

$$\leq 4l^2 (t-1) \nu(x^{(t)}) - a_t \cdot (\bar{v}_t - a_t)$$

Then sum both sides of the inequality for $t = 1, \ldots, n$ with the left-hand side telescoping to become $n^2 \|\bar{v}_{n+1} - a_{n+1}\|^2$. 


Dividing both sides by \( n^2 \),
\[
\norm{\tau_{n+1} - a_{n+1}}^2 \\
\leq \frac{4}{n} + \frac{2}{n} \sum_{i=1}^{n} \left( \frac{t-1}{n} \right) (v(x^{(i)})) - a_t) \cdot (\tau_t - a_t) \\
\leq \frac{4}{n} + \frac{2}{n} \sum_{i=1}^{n} \left( \frac{t-1}{n} \right) \beta_i (v(x^{(i)})) - a_t) \cdot \norm{\tau_t - a_t} \\
= \frac{4}{n} \tag{24}
\]
Because \( \frac{t-1}{n} \norm{\tau_t - a_t} \in [0, 2] \), \( \beta_t - a_t \geq 0 \) component-wise, and by construction of \( x_t \),
\[
\norm{\tau_{n+1} - a_{n+1}}^2 \leq \frac{4}{n} + \frac{8}{n} \sum_{i=1}^{n} \beta_i (v(x^{(i)})) - a_t) \\
\leq \frac{4}{n} + \frac{8}{n} \sum_{i=1}^{n} \beta_i (v(x^{(i)})) \\
\leq \frac{4}{n} \tag{25}
\]
By triangle inequality, after \( n \) steps, we have a \( \frac{4}{n} \)-CCE \( x \) with \( \mathcal{C}(x) \geq y - \frac{4}{n} \). Substituting \( \epsilon = \frac{4}{n} \), we have a \( \text{Poly}(\frac{1}{\epsilon}, n, m) \) runtime.

\[\text{A.2. Proof of Lemma 2}\]

**Definition A.1.** A cost-minimization game is defined as \((\lambda, \mu)\)-smooth if, given cost functions \( c_i \),
\[
\sum_{i=1}^{N} c_i(x_i^*, x_{-i}) \leq \lambda \sum_{i=1}^{N} c_i(x^*) + \mu \sum_{i=1}^{N} c_i(x), \tag{26}
\]
for all mixed strategies \( x_i^*, x^* \).

The expression \( \frac{\lambda}{\mu} \) is known as the robust price-of-anarchy (POA) ([Roughgarden, 2015](#)).

**Lemma 2.** Let \( \rho \) denote an upper bound on the robust price of anarchy of \( G, \tilde{G} \). There exists a \( \text{Poly}(1/\epsilon, n, m, \rho) \)-time halfspace oracle that, given a vector \( \beta \in \mathbb{R}^{1+m} \) and \( y \leq \rho \max_{x \in \mathcal{CCE}} \mathcal{C}(x) \), returns an \( x \in P(A) \) such that \( \beta v(x) \leq \epsilon \) where \( v \) is defined as in Equation 8.

**Proof.** We will prove that this oracle exists by construction. Limiting candidate \( x \) to product distributions, the oracle is equivalent to solving an optimization problem over \( m \) variables: \( x_1^*, \ldots, x_m^*, \ldots, x_N^* \) where \( x_i^* \) corresponds to the probability that agent \( i \) plays action \( j \) under \( x \).

The objective is minimizing
\[
\beta v(x) = -\beta_{m+1} (\mathcal{C}(x) - y) \\
- \sum_{i=1}^{N} \left( \sum_{j=1}^{m_i} \beta_j \left( \pi_i(x) - \sum_{a_{-i}} \frac{a_{-i} u_i(j, a_{-i})}{\beta_j} \right) \right) \\
= -\beta_{m+1} (\mathcal{C}(x) - y) \\
- \sum_{i=1}^{N} \beta_i \left( \pi_i(x) - \pi_i \left( \frac{\beta_i}{\beta_i \|x_i\|_1}, x_{-i} \right) \right) \tag{27}
\]
down to a non-positive value. For convenience, we’ll write,
\[
\mathcal{C}(x) = -\beta_{m+1} (\mathcal{C}(x) - y) \\
- \sum_{i=1}^{N} \beta_i \left( \pi_i(x) - \pi_i \left( \frac{\beta_i}{\beta_i \|x_i\|_1}, x_{-i} \right) \right) \tag{28}
\]
Note that efficiently finding a \( x \) with a guaranteed negative upper bound lets us specify that upper bound as a condition \( y \).

This optimization problem is combinatorial and generally intractable. However, we can exploit the usually high social welfare of no-regret learning. Consider no-regret learning with \( n \) agents that have the cost function,
\[
c_i(x) = -\beta_{m+1} (\mathcal{C}(x) - y) - \beta_i \left( \pi_i(x) - \pi_i \left( \frac{\beta_i}{\beta_i \|x_i\|_1}, x_{-i} \right) \right), \tag{29}
\]
with action space \( A_i \). Let \((\lambda, \mu)\) be the smoothness of this game.

Take \( x \) as our average mixed strategy under \( t \) no-regret dynamics. Since per agent, by standard no-regret guarantees, for any \( x_i \in A_i \),
\[
c_i(x) = -\beta_{m+1} (\mathcal{C}(x) - y) - \beta_i \left( \pi_i(x) - \pi_i \left( \frac{\beta_i}{\beta_i \|x_i\|_1}, x_{-i} \right) \right), \tag{30}
\]
we can sum over all agents to apply our smoothness properties.

\[
\mathcal{C}(x) = \sum_{i=1}^{n} c_i(x) \tag{31}
\]
\[
\leq \sum_{i=1}^{n} c_i(x_i^*, x_{-i}) \tag{32}
\]
\[
\leq \lambda \mathcal{C}(x^*) + \mu \mathcal{C}(x) \tag{33}
\]
We know there is always a mixed strategy \( x^* \) that guarantees \( \mathcal{C}(x^*) \leq -\max_{x \in \mathcal{CCE}} \mathcal{C}(x) \). This implies that, taking for granted that \( 1 - \mu \neq 0 \),
\[
(1 - \mu) \mathcal{C}(x) \leq \lambda \mathcal{C}(x^*) \tag{34}
\]
\[
\mathcal{C}(x) \leq -\frac{\lambda \max_{x \in \mathcal{CCE}} \mathcal{C}(x)}{1 - \mu} \tag{35}
\]
Recall we require that $C(x) + \beta_{m+1} y = \beta v(x) \leq 0$, meaning we must choose
\[
\frac{\lambda \max_{x \in \text{CCE}} \mathcal{P}(x)}{\mu - 1} + \beta_{m+1} y \leq 0
\]
\[\Rightarrow y \leq \frac{\lambda \max_{x \in \text{CCE}} \mathcal{P}(x)}{\beta_{m+1}(1 - \mu)} \leq \frac{\lambda \max_{x \in \text{CCE}} \mathcal{P}(x)}{(1 - \mu)}
\]
(37)

We now seek to lower bound $\frac{\lambda}{1 - \mu}$. First, note that $c_i$ is composed of three additive terms:
\[
c_i^{(1)}(x) = -\beta_{m+1} \mathcal{P}(x)
\]
(38)
\[
c_i^{(2)}(x) = -\sum_{i=1}^{N} \|\beta_i\|_1 \pi_i(x^*)
\]
(39)
\[
c_i^{(3)}(x) = +\sum_{i=1}^{N} \|\beta_i\|_1 \pi_i \left( \frac{\beta_i}{\|\beta_i\|_1}, x_{-i}^* \right)
\]
(40)

First note that the third term $c_i^{(3)}$ does not contribute any meaningful restrictions on smoothness. Take any mixed strategies $x, x^*$,
\[
\sum_{i=1}^{N} c_i^{(3)}(x^*_i, x_{-i}) = \sum_{i=1}^{N} \|\beta_i\|_1 \left( \pi_i \left( \frac{\beta_i}{\|\beta_i\|_1}, x_{-i}^* \right) \right),
\]
(41)
\[
\sum_{i=1}^{N} c_i^{(3)}(x) = \sum_{i=1}^{N} \|\beta_i\|_1 \left( \pi_i \left( \frac{\beta_i}{\|\beta_i\|_1}, x_{-i} \right) \right)
\]
(42)

Thus, we can bound our robust POA with that of $\frac{\lambda \frac{\lambda}{1 - \mu}}{1 - \mu}$ and $\frac{\lambda \frac{\lambda}{1 - \mu}}{1 - \mu}$.

**A.4. Proof of Proposition 1**

We first motivate our study of how behavioral deviations affect the generalization of strategic policies. Specifically, we will look at examples of where naive strategic decision-making can result in significant regret when confronted with boundedly rational agents—even when said agents are asymptotically rational.

For simplicity, we’ll discuss a traditional single-follower Stackelberg setting where $N = 1$. In this setting, we’ll refer to agent 0 as the leader and to agent 1 as the follower. Recall that in fully rational settings, we can anticipate the follower to play a best response to whatever mixed strategy the leader adopts: $X = \text{BR}(x_0) := \arg\max_{a_1 \in A_1} \mathbb{E}_{x_0 \sim \pi_0}[u_1(a_0, a_1)]$.

Consider an instance of bounded rationality where the leader plays a strategy $x_0$ but the follower perceives the leader as having committed to the strategy $\tilde{x}_0$. This may arise in partial observability settings or when the follower is limited to bandit feedback, for example. A classical statement of the fragility of Stackelberg equilibria is that even when the follower’s error, $\|x_0 - \tilde{x}_0\|$, is infinitesimally small, the follower’s bounded rationality can cost the leader a constant value bounded away from zero.

**Proposition 2.** Fix a leader mixed strategy $x_0 \in P(A_0)$. There exists Stackelberg game $(N = 1)$ with bounded payoffs where there is a sequence of mixed strategies $x_0(1), x_0(2), \cdots \in P(A_0)$ such that $\lim_{t \to \infty} \|x_0 - x_0(t)\| = 0$ but for all $t$, $\min_{a_1 \in \text{BR}(x_0)} u_0(x_0, x_1) - \max_{x_1 \in \text{BR}(x_0(1))} u_0(x_0, x_1)$ is positive and bounded away from zero.

These situations arise when the relationship between the leader and follower’s utility functions is not smooth. Samuelson’s game is an example of where this may result in non-robust optimal strategies independent of a Stackelberg assumption. Consider the below payoff matrix.

|       | L  | R  |
|-------|----|----|
| T     | 100| 50 |
| B     | 99 | 99 |

There is a unique Nash equilibria at $a_0 = T, a_1 = L$. Accordingly, optimizing against the best response of the column-player would recommend a $T$ pure strategy to the row-player. However, if the column-player is boundedly rational or otherwise acts according to a perturbed payoff matrix, a safer strategy for the row-player is to play the action $B$, which guarantees at least a payoff of 99. Optimizing against the worst-case $B$-best response of the column-player, where $\epsilon \geq 1$, would recommend this robust $B$ strategy to the row-player. Extending this insight into the $N > 1$ domain, we similarly want to optimize against worst-case equilibria to afford robustness against payoff structures such as...
Proposition 1. The uncertainty set $X'$ of (any combination) of agents violating assumptions 1-3 with parameters $\gamma_m, \gamma_s, \gamma_p$ is contained in the set of $\epsilon$-CCE:

$$CCE_{\epsilon} = \{ x \in P(A) \mid \forall i \in [1, \ldots, N], a_i \in A_i : \mathbb{E}_{a \sim \xi}[u_i(a)] + \epsilon_i \geq \mathbb{E}_{a \sim \xi}[u_i(\hat{a}_i, a_{-i})] \},$$

where $\epsilon_i = \max\{ \frac{|u_i|_\infty}{1 - \gamma_m}, \gamma_s, 2\gamma_p \}$. Hence, we can train policies robust to such agents by using:

$$X_{\epsilon}(x_0) = \{ x \in P(A) \mid \exists x_0 \in A_0 : (x_0, x) \in CCE_{\epsilon} \}$$

in the robustness objective of Equation 1.

The proof of this proposition follows immediately from the observation that all three uncertainties—subjective rationality, procedural rationality, and myopicness/commitment power—can be framed as uncertainty about agent utility functions.

Proof. First, subjective rationality is by definition an uncertainty about agent utility functions. Our parameterization of this uncertainty yields a natural uncertainty set over reward functions. Recall that we upper bound the infinity norm by $\gamma_s$, which is defined as the inf $\gamma$ s.t. $|u_i - \hat{u}_i|_\infty := \max_{a \in A} |u_i(a) - \hat{u}_i(a)| \leq \gamma$. We accordingly define the uncertainty set $\Xi_{\epsilon}(u_i) = \{ u_i : A \rightarrow \mathbb{R} \mid |u_i - \hat{u}_i|_\infty \leq \epsilon \}$.

Similarly, we can define an uncertainty set around non-myopic agents and general uncertainty about our commitment power by upper-bounding possible perturbations to an agent’s originally anticipated utility function. In particular, $|u_i - (u_i(x) + \sum_{t=1}^{\infty} \gamma_m u_t(x_i))| \leq \frac{|u_i|_\infty}{1 - \gamma_m} = \epsilon$. Hence, all non-myopic agents with $\gamma_m \leq 1 - \frac{|u_i|_\infty}{\epsilon}$ will be expressed in the uncertainty set $\Xi_{\epsilon}(u_i)$.

Procedural rationality can also be framed in terms of $\Xi_{\epsilon}(u_i)$. In particular, the $\epsilon$-coarse correlated equilibria under $u_i$ exactly coincides with the union of coarse correlated equilibria under reward perturbations of up to $\epsilon/2$. Formally, $x$ is in an $\epsilon$-coarse correlated equilibria under $u_i$ if and only if,

$$\forall i \in [1, \ldots, N], x_i \in P(A_i) : \pi_i(x) + \epsilon_i \geq \pi_i(x_i, x_{-i})$$

$x$ is in a coarse correlated equilibria under some $\overline{u}'_i \in \Xi_{\epsilon}(u_i)$ if and only if,

$$\exists \overline{u}'_i \in \Xi_{\epsilon}(u_i), s.t. \forall i \in [1, \ldots, N], x_i \in P(A_i) : \overline{u}'_i(x) \geq \overline{u}'_i(x_i, x_{-i}),$$

which is equivalent to the condition,

$$\forall i \in [1, \ldots, N], x_i \in P(A_i) : \overline{u}'_i(x) + \epsilon \geq \overline{u}'_i(x_i, x_{-i}) - \epsilon,$$

Hence, the set of $\frac{\epsilon}{2}$-equilibria behaviors, up to $(1 - \frac{|u_i|_\infty}{\epsilon})$-discount factor non-myopic agent behaviors, and up to $\epsilon$ reward perturbation-consistent behaviors are all contained in the set of $\epsilon$-CCE.

B. Appendix: Additional Experiments

B.1. Ablation Study: Dynamic vs Fixed Lagrange multipliers $\lambda$

In our robust learning framework, Algorithm 1, the Lagrange multipliers $\lambda$ play an important role in moderating the self-play dynamics used to sample adversarial dynamics. Specifically, the multiplier $\lambda_i$ for agent $i$ balances agent $i$’s incentive to improve its own reward with its obligation to act adversarially to the ego agent by minimizing the ego agent’s reward. Recall that smaller values of $\lambda$ yield more antagonistic agent objectives. As described in Algorithm 4, these multipliers $\lambda$ are periodically updated using local Monte-Carlo estimates of regret. This raises the question of whether we can instead fix a constant non-zero value for the multipliers $\lambda$ and still retain an effective adversarial equilibria sampler. This experiment answers that question in the negative. Figure 5 visualizes the equilibria discovered with a fixed $\lambda_0$ in the same format as in Figure 2 (middle). This comparison shows that using a fixed $\lambda_0$ affects the equilibria discovered by our algorithm; the bottom right quadrant which contains the $\epsilon$-equilibria discovered with the dynamic multipliers $\lambda$ used by Algorithm 4 are not reached for any values of frozen multipliers $\lambda_0$. This demonstrates that, even in simple game settings, certain $\epsilon$-equilibria are only reachable with dynamic $\lambda$ and hence multiplier updates are necessary for proper behavior.

Figure 5. Using fixed values of $\lambda$ (rather than allowing it to update, as in the full algorithm) distorts performance and prevents agents from reaching the same $\epsilon$-equilibria discovered with learned $\lambda$. 

---

Learning Adversarially Robust Policies in Multi-Agent Games
B.2. Additional Experiment Details

The hyperparameters for our deep dynamic mechanism design experiments are listed in Table 1 for the N-Matrix games (depicted in Figure 3), and Table 2 (depicted in Figure 4) for the AI Economist tax policy game. Additional hyperparameters specific to the AI Economist simulation environment were kept at default values as described in the manuscript (Zheng et al., 2020).

Table 1 details: In the N-Matrix game experiments, we train 20 models for each training method and conduct evaluation runs of each model in every test environment. However, evaluation runs may fail to stabilize. Thus, we drop the bottom 10 evaluation runs for each training method.

Table 2 details: In the AI Economist tax policy experiments, we train 9 models for each training method and conduct evaluation runs of each model in every test environment. However, evaluation runs may fail to stabilize. Thus, we drop the bottom 6 evaluation runs for each training method. Furthermore, for this complex task, the adversary in Algorithm 4 may fail to converge in reasonable time. Thus, for the Algorithm 4 training method, we select its 9 trained models from a set of 15 candidate models (i.e., dropping at most 6 failed training runs). This selection is made programmatically on the basis of each model’s median reward on validation runs in a standalone validation environment ($\eta = 0.27$). We similarly choose the $\epsilon = 30$ hyperparameter for Algorithm 3 from a grid-search over $\epsilon \in \{0, -10, -20, -30, -40\}$ on the basis of validation reward.
Table 2. Hyperparameters for AI Economist Experiments

| Parameter                           | Value          |
|-------------------------------------|----------------|
| Training algorithm                  | PPO            |
| Pretrained Episodes                  | 450,000        |
| Finetuned Episodes                   | 20,000         |
| CPUs (Baselines)                    | 15             |
| CPUs (Algorithm 4)                  | 95             |
| Number of agents                    | 4              |
| Episode length                      | 1000           |
| World dimensions                    | 25x25          |
| Default iso-elastic $\eta$          | 0.23           |
| Training Seeds                      | 9              |
| Test Seeds                          | 3              |
| Number of convolutional layers      | 2              |
| Number of fully-connected layers    | 2              |
| Fully-connected layer dimension (agent) | 128          |
| Fully-connected layer dimension (planner) | 256         |
| LSTM cell size (agent)              | 128            |
| LSTM cell size (planner)            | 256            |
| All agents share weights            | True           |
| Learning Rate (Ego Agent)           | 0.0001         |
| Learning Rate, $\eta$ (Other Agents) | 0.0003       |
| Entropy regularization (Ego Agent)  | 0.1            |
| Entropy regularization, $\alpha$ (Other Agents) | 0.025         |
| Gamma                               | 0.998          |
| GAE Lambda                          | 0.98           |
| Gradient clipping                   | 10             |
| Value function loss coefficient     | 0.05           |
| SGD Minibatch Size                  | 3000           |
| SGD Sequence Length                 | 50             |
| Value/Policy networks share weights | False          |

PPO Parameters

C. Appendix: Additional Related Work

In this section, we discuss additional related work regarding adversarially robust reinforcement learning. The topic of adversarially robust reinforcement learning originates from control theory literature. Morimoto & Doya (2001) proposed a form of adversarially robust reinforcement learning, inspired by $H_{\infty}$ control, for attaining robustness to uncertainty about the environment dynamics. Later, Pinto et al. (2017) proposes to learn an adversary policy to induce this robustness, again targeting robustness to uncertainty about environment dynamics in a control-theoretic sense.

Other works have since studied perturbing transition matrices, observation spaces, and action probabilities of Markov decision processes, with motivating applications in robotics and control theory (Tessler et al., 2019; Hou et al., 2020). These concepts have recently been extended to multi-agent settings, such as by (Li et al., 2019). However, these prior work on robust multi-agent learning differ from ours in two important ways. First, their multi-agent task is controlling a swarm (of multiple agents) in an efficient and coordinated fashion. Our multi-agent task is designing a policy that will yield favorable strategic outcomes in a multi-agent game. Second, their works analyze control-theoretic notions of robustness, extending single-agent policy techniques to a multi-agent policy setting.

D. Appendix: Algorithm Pseudocodes

In this section, we provide the full pseudocode description of Algorithms 2, 3, and 4.

Algorithm 2 Method of adversarially robust mechanism design.

Output: Adversarially robust policy $x_0$.

Input: Simulation environment $S$, robustness hyperparameter $\epsilon$, $n$ training steps, bandit algorithm $B$.

Initialize policy $x_0$ and a bandit algorithm.

for $i = 1, \ldots, n$ do

Query algorithm $B$ for a mixed strategy $x_0$, and sample an action $a_0$.

Run Algorithm 1 or 3 in $S$ to approximate a lower bound $\ell \leq \min_{\pi \in \mathcal{CCE}} \mathbb{E}_{A \sim \pi} [u_0(a_0, \pi)]$.

Pass $\ell$ as a reward to bandit algorithm $B$.

end for

Return the average strategy played by $B$. 
Algorithm 3 Poly\((n, m, \rho, \varepsilon)\)-time algorithm for sampling approximately optimal CCE.

**Output:** $\varepsilon$-CCE $x$ such that

\[
\pi(x_0, x) \geq \frac{1}{\rho} \max_{\pi \in \pi_{\varepsilon-CCE}} \pi(x_0, \bar{x}) - \varepsilon.
\] (45)

**Input:** Generalization parameter $\varepsilon \in \mathbb{R}^n$, tolerance $\varepsilon$, objective function $\nu : A_0 \times A \rightarrow \mathbb{R}$, ego agent policy $x_0$, bandit feedback access to utility functions $\bar{u}_0, \ldots, \bar{u}_N$.

while Binary search over $y \in \mathbb{R}$ (within $\nu$’s payoff bounds) is coarser than the desired Poly\(|I|\) resolution do

Initialize some $x \in P_{\text{prod}}(A)$.

for $O\left(\frac{1}{\varepsilon^2}\right)$ iterations indexed by $t$ do

Compute the vector $v_i$ from Theorem 1 (Eq 8).

Compute importance weight vector,

\[
\beta_t \leftarrow \begin{bmatrix}
\max\{v_1 - \varepsilon_1, 0\} \\
\vdots \\
\max\{v_m - \varepsilon_1, 0\} \\
\vdots \\
\max\{v_{m-1} - \varepsilon_N, 0\} \\
\vdots \\
\max\{v_m - \varepsilon_N, 0\}
\end{bmatrix}.
\] (46)

Normalize $\beta_t \leftarrow \beta_t / \|\beta_t\|_1$.

Run $O\left(\frac{1}{\varepsilon^2}\right)$ steps of simultaneous deterministic no-regret dynamics on $n$ agents with cost function,

\[
e_t(x) = -\frac{\beta_{m+1}}{n} \pi(x) - \|\beta_t\|_1 \left( \pi_t(x) - \pi_t \left( \frac{\beta_t}{\|\beta_t\|_1}, x_{-i} \right) \right),
\] (48)

where $\beta_i$ are the components of $\beta$ starting with index $1 + \sum_{j=1}^{i-1} m_j$ and ending with index $\sum_{j=1}^{i} m_j$ inclusive.

Store empirical play distribution of the no-regret dynamics as $x_t$. If $\beta_t x_t \geq \varepsilon$, $y$ is too large. Terminate loop and resume binary search.

end for

If the prior loop was successful ($y$ was not too large), store the average empirical play distribution over $x_1, x_2, \ldots, x^*$. Return $x^*$.

end while

Return $x^*$.

Algorithm 4 Adversarially robust dynamic mechanism design.

**Output:** Learned parameters, $\theta_0$, for a mechanism represented as agent 0.

**Input:** Reward slack $\varepsilon$, learning rate $\lambda$, batch size $\beta$, initial agent parameters $\theta = [\theta_1, \ldots, \theta_N]$.

**Input:** Number of rounds $n_{\text{rounds}}$, evaluation batches $n_{\text{eval}}$, training batches $n_{\text{train}}$.

for $j = 1, \ldots, n_{\text{rounds}}$ do

Copy $\theta$ into placeholders $\tilde{\theta} \leftarrow \theta$.

for $j = 1, \ldots, n_{\text{eval}}$ do

Accumulate $\beta$ timesteps under $\tilde{\theta}, \theta_0$.

For each agent $i \geq 1$, store their experience in a tuple of lists,

\[
B_{i,j} := (\text{Rew}_{i,j}, \text{Obs}_{i,j}, \text{Actions}_{i,j}).
\]

Update each agent $i \geq 1$ critic (if used) and actor for $\tilde{\theta}$ with $B_i$.

end for

Update each agent $i \geq 1$ multiplier:

\[
\lambda_i \leftarrow \lambda_i - \alpha \lambda \text{Mean} (\text{Rew}_{i,n_{\text{train}}}, \varepsilon - \text{Rew}_{i,0})
\]

for $j = 1, \ldots, n_{\text{train}}$ do

Accumulate $\beta$ timesteps under $\theta$ and $\theta_0$ and store them in $B_{0,j}, \ldots, B_{N,j}$.

Update the mechanism’s $\theta_0$ critic and actor with $B_0$ using a slow learning rate.

Update each agent $i \geq 1$ critic for $\theta$ (if used) with $B_i$.

Update each agent $i \geq 1$ actor for $\theta$ with,

\[
\tilde{B}_{i,j} = (\text{Rew}_{i,j}, \text{Obs}_{i,j}, \text{Actions}_{i,j})
\]

where the recorded reward of agent $i$, at each timestep $t = 1, \ldots, \beta$, is modified:

\[
\tilde{\text{Rew}}_{i,j} = \left[ \frac{\lambda_t \text{Rew}_{i,t}^{(j)} - \text{Rew}_{0,j}^{(t)}}{1 + \lambda_i} \right] | t = 1, \ldots, \beta
\]

end for

end for

Return parameters $\theta_0$. 
