Learning to Play General-Sum Games against Multiple Boundedly Rational Agents

Eric Zhao[1,2], **Alexander R. Trott**[3], **Caiming Xiong**[1], **Stephan Zheng**[1]

[1]Salesforce Research. Palo Alto, California, USA
[2]University of California, Berkeley. Berkeley, California, USA
[3]MosaicML. San Francisco, California, USA

Abstract

We study the problem of training a principal in a multi-agent general-sum game using reinforcement learning (RL). Learning a robust principal policy requires anticipating the worst possible strategic responses of other agents, which is generally NP-hard. However, we show that no-regret dynamics can identify these worst-case responses in poly-time in smooth games. We propose a framework that uses this policy evaluation method for efficiently learning a robust principal policy using RL. This framework can be extended to provide robustness to boundedly rational agents too. Our motivating application is automated mechanism design: we empirically demonstrate our framework learns robust mechanisms in both matrix games and complex spatiotemporal games. In particular, we learn a dynamic tax policy that improves the welfare of a simulated trade-and-barter economy by 15%, even when facing previously unseen boundedly rational RL taxpayers.

Introduction

We study the problem of learning a principal policy in a general-sum game against boundedly rational agents. Learning a robust principal policy requires us to anticipate how these agents may respond to our policy choices, and entails two important challenges (Figure 1). First, the policies we choose induce a sub-game between the other agents, a sub-game which may have infinite equilibria. The policy we choose should perform well regardless of which equilibria the agents respond with. Second, principal policies that perform well against rational agents may not generalize to boundedly rational agents, even if they are only infinitesimally irrational (Pita et al. 2010). Our policy should perform well even if agents act boundedly rational.

We introduce a framework for the reinforcement learning of robust principal policies that address these two challenges. This framework evaluates a potential policy by identifying the worst-case coarse-correlated equilibrium (CCE) of the sub-game the policy induces. Although identifying *worst-case* CCE is generally computationally intractable (Papadimitriou and Roughgarden 2008; Barman and Ligett 2015), we prove that worst-case CCE can efficiently learned in smooth games. Our framework easily extends to identify worst-case approximate CCE. This allows us to learn principal policies that are robust to boundedly rational agents, such as agents whose incentives differ from the agents we train against.

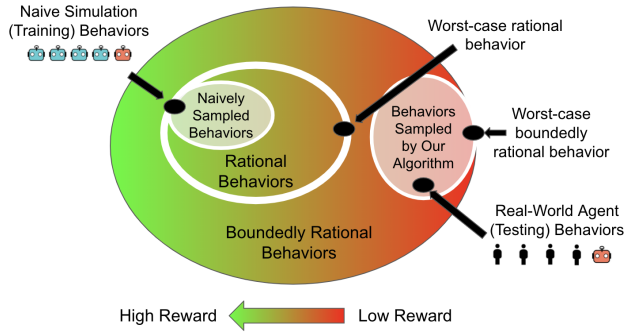

Figure 1: The orange robot denotes our principal, blue robots the agents we train against, and human icons the agents we encounter during test time. Evaluating a policy in a multi-agent game by naively sampling rational responses from other agents, e.g. via multi-agent RL, may lead to overly optimistic reward estimates. We introduce efficient algorithms for adversarially sampling rational responses in smooth games. These algorithms can be extended to sample worst-case boundedly rational responses (bottom-right).

Our motivating application is *mechanism design* ("reverse game theory"), where a principal 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 equilibria behavior of agents can be solved in closed-form. Recent research have pursued computational approaches to mechanism design that evaluate potential mechanisms using agent-based modeling (Holland and Miller 1991; Bonabeau 2002; Duetting et al. 2019) and multi-agent reinforcement learning (MARL) (Zheng et al. 2020). This application of multi-agent learning remains an exciting but understudied topic. We will outline a modern perspective that formalizes automated mechanism design and its robustness concerns as an equilibrium selection problem.

Summary of results. Our primary contributions include:

1. *We motivate a robust learning objective* for finding a principal policy that is robust to agents of differing incentives, rationality, and reputation. This objective is a multi-follower extension of robust Stackelberg games.
2. *We show the existence of poly-time algorithms* for adversarially sampling the coarse-correlated-equilibria (CCE) of smooth games, proving that multi-follower Stackelberg games can be tractable. This weakens prior findings that learning welfare-maximizing CCE is NP-hard (Papadimitriou and Roughgarden 2008).
3. *We apply our proposed framework to automated mechanism design* problems where multi-agent RL is used to simulate the outcomes of mechanisms. In the spatiotemporal economic simulations used by the AI Economist (Zheng et al. 2020), our framework learns robust tax policies that improve welfare by up to 15% and are robust to previously unseen and boundedly rational agents.[1]

Related Work

Finding Equilibria. A goal of multi-agent learning is finding equilibria (or more generally *solution concepts*), i.e., sets of agent 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, Zheng, and Anandkumar 2020). However, learning *robust* principals (or mechanisms) in multi-agent general-sum games is an open problem: it requires evaluating strategies against worst-case sub-game equilibria, which is computationally hard.

Stackelberg Games. Our principal can be seen as a Stackelberg *leader*; the other agents as Stackelberg *followers*. Stackelberg games have had real-world success in security games used in airports (Pita et al. 2009) and anti-poaching defense (Nguyen et al. 2016), for example. However, Stackelberg analysis is typically limited to a single rational follower or assuming that the followers do not strategically interact, e.g., as in multi-follower security games (Korzhyk, Conitzer, and Parr 2011). In contrast, we consider more general settings with multiple followers who may strategically interact with one another. In these settings, finding multi-follower "best-responses" is a computationally hard equilibria search and an open problem (Barman and Ligett 2015; Basilico et al. 2020) for which our work provides a tractable perspective.

Our approach to modeling uncertainty about followers is inspired by *Strict Uncertainty Stackelberg Games* that assume a worst-case choice of a follower's utility function (Pita et al. 2010, 2012; Kiekintveld, Islam, and Kreinovich 2013). Similarly, *Bayesian Stackelberg Games* assume a prior over a space of possible follower utility functions (Conitzer and Sandholm 2006). We extend this by considering a general setting with multiple, possibly interacting, followers.

Robustness. Our work is also related in spirit to prior work on robust reinforcement learning, which we discuss further in Appendix . We will refer to our notion of robustness as *strategic robustness* to contrast it from non-game-theoretic notions of robustness to, for example, noisy observations (Morimoto and Doya 2000). Strategic robustness also differs from the topic of "robust game theory", which studies the equilibria that arise when all players act robustly to some uncertainty about game parameters (Aghassi and Bertsimas 2006). Hereafter, we will simply refer to our notion of strategic robustness as "robustness" for brevity.

Problem Formulation

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

Setup. We consider a general-sum game G with $N+1$ agents. Our principal, or "ego agent", is index $i = 0$ and the other agents are $i = 1, \ldots, N$. A_i denotes the set of m_i actions available to agent i, and $m = \sum_{i=1}^{N} m_i$. $P(A_i)$ is the set of probability distributions over action set A_i. The *joint action set* is $A_{1:n} := \prod_{i \in [1,\ldots,N]} A_i$. $P(A_{1:n})$ denotes the set of joint distributions over strategy profiles $A_{1:n}$. $P^{\text{prod}}(A_{1:n})$ denotes the set of product distributions over strategy profiles $A_{1:n}$. Every agent $i = 0, \ldots, N$ has a utility function $u_i : A_0 \times A_{1:n} \to \mathbb{R}$ with bounded payoffs. For example, $u_i(a_0, \boldsymbol{a})$ denotes the utility of agent i under action $a_0 \in A_0$ by our principal and actions $\boldsymbol{a} \in A$ by agents $1, \ldots, N$. When a_0 is clear from context, we'll write $u_i(\boldsymbol{a})$, suppressing a_0. We denote expected utility as $\overline{u}_i(x_0, \boldsymbol{x}) := \mathbb{E}_{a_0 \sim x_0, \boldsymbol{a} \sim \boldsymbol{x}} u_i(a_0, \boldsymbol{a})$; again we write $\overline{u}_i(\boldsymbol{x})$ 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 and Roughgarden 2008). We assume $G := (I, T, U)$ is a succinct game, i.e., has a polynomial-size string representation. 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 of *polynomial type*, i.e., $n, m_0, \ldots, m_N$ are polynomial bounded in $|I|$. We assume that G satisfies the *polynomial expectation property*, i.e., utilities $\overline{u}(x_0, \boldsymbol{x})$ can be computed in polynomial time for product distributions $\boldsymbol{x}$. The latter assumption is known to hold for virtually all succinct games of polynomial type (Papadimitriou and Roughgarden 2008). Without these assumptions, simply evaluating the payoff of a coarse-correlated equilibria can require exponential time. All complexity results in our work, including Theorem 1 and cited results from prior works use these assumptions. Later, we will use additional "smoothness" conditions on G to overcome prior hardness results about succinct games.

[1] Source code for these experiments are released at https://github.com/salesforce/strategically-robust-ai.

Problem. Given a game G, we aim to learn a principal policy $x_0 \in P(A_i)$ that maximizes our principal's expected utility $\overline{u}_0(x_0, \boldsymbol{x})$. Here, $\boldsymbol{x}$ is the strategy that agents in a test environment respond to our policy x_0 with. We will assume access to a training environment for G. In this training environment, we assume access to—potentially inaccurate—estimates of the reward functions of the agents; we will write these estimates as $u_1, \ldots, u_N$. For example, the principal $i = 0$ may be a policymaker setting a tax policy x_0 that maximizes social welfare u_0. In response, the tax-payers play $\boldsymbol{x}$, choosing whether to work and report income.

Objective In order to formalize our robust learning objective, we must define what uncertainty sets $\boldsymbol{X}$ we want learning guarantees for. A behavioral uncertainty set $\boldsymbol{X}(x_0) \subseteq P(A)$ defines the strategies that the test environment agents may respond to a principal policy x_0 with. For simple agent behaviors, one can use imitation learning or domain knowledge to construct these uncertainty sets. In this work, we will study game-theoretic uncertainty sets, for example, where $\boldsymbol{X}(x_0)$ is the set of rational behaviors (see Section). Fixing a choice of $\boldsymbol{X}$, we can write our robust learning objective as:

$$\max_{x_0} \min_{\boldsymbol{x} \in \boldsymbol{X}(x_0)} \overline{u}_0(x_0, \boldsymbol{x}). \tag{1}$$

This is a challenging objective: it features nested optimization and requires searching over behavioral uncertainty sets, which is a non-trivial task in complicated games.

Finding Uncertainty Sets for Boundedly Rational Agents

A key challenge in our problem formulation is defining our behavioral uncertainty set $\boldsymbol{X}$. In this section, we will first argue for a uncertainty set of *coarse-correlated equilibria (CCE)*. We will then prove that in *smooth games* we can efficiently approximate worst-case CCE. We will finally propose a more relaxed uncertainty set that yields robustness to boundedly rational agents.

Initial Assumptions Our guiding principle for choosing $\boldsymbol{X}$ is that a robust principal can assume that agents are rational, but should still perform if agents act somewhat irrationally or have incentives that slightly differ from anticipated. We will further assume the following and relax them later.

1. The incentives (reward functions) of the test agents exactly agree with our training environment's estimates.
2. All agents are rational expected-utility maximizers. No agent will settle if they want to unilaterally deviate.
3. We, the principal, can commit to a strategy x_0 and will not react to other agents.

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

$$\boldsymbol{X}(x_0) = \{\boldsymbol{x} \mid \forall i \in [1, \ldots, n], \tilde{\boldsymbol{x}}_{-i} \in P(A)_{-i} : x_i \in \text{argmax}_{x_i} \overline{u}_i(x_0, x_i, \tilde{\boldsymbol{x}}_{-i})\}. \tag{2}$$

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.

Stability-Based Equilibria. We can also define an uncertainty set $\boldsymbol{X}(x_0)$ as the stable equilibria that agents may converge to under our policy x_0. This coincides with Equation 2 when it is non-empty. Formally, let

$$\boldsymbol{X}(x_0) = \{\boldsymbol{x} \in P(A) \mid (x_0, \boldsymbol{x}) \in \text{EQ}\},$$

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

$$\text{MNE} = \{\boldsymbol{x} \in P^{\text{prod}}(A) \mid \forall i \in [1, \ldots, N], \tilde{a}_i \in A_i : \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{x}}[u_i(\boldsymbol{a})] \geq \mathbb{E}_{\boldsymbol{a}_{-i} \sim \boldsymbol{x}_{-i}}[u_i(\tilde{a}_i, \boldsymbol{a}_{-i})]\},$$

or more general coarse-correlated equilibria (CCE):

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

Here, coarse-correlated equilibria describe more general joint strategies, 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 the MNE/CCE that minimizes a utility function u_0 is equivalent to the NP-hard problem of finding a MNE/CCE that maximizes a linear social welfare objective ν (Daskalakis, Goldberg, and Papadimitriou 2009; Papadimitriou and Roughgarden 2008); we will set $\nu = -u_0$ for convenience. Beyond maximizing ν, simply finding a CCE that does not minimize ν is NP-hard (Barman and Ligett 2015). Formally, consider the decision problem Γ of determining whether a game G (under our assumptions) admits a CCE $\boldsymbol{x}$ such that the expectation of ν, $\overline{\nu}$, satisfies:

$$\overline{\nu}(\boldsymbol{x}) > \min_{\tilde{\boldsymbol{x}} \in \text{CCE}} \overline{\nu}(\tilde{\boldsymbol{x}}).$$

This problem is NP-hard for some choices of ν, including the social welfare function (Barman and Ligett 2015). For our purposes, this implies that even sampling an approximately worst-case equilibria is intractable. This means it could be impossible to efficiently evaluate our principal's policy as it is intractable to guarantee sampling anything other than uninformative equilibria behavior.

Smooth Games and Tractable Uncertainty Sets

Smooth games offer a workaround to this hardness result.

Definition 0.1 (Smooth Games)**.** A *cost-minimization game* with cost functions c_i and objective C is (λ, μ)-smooth if, for all strategies $\boldsymbol{x}, \boldsymbol{x}^* \in P(A)$,

$$\sum_{i=1}^{N} \mathbb{E}[c_i(x_i^*, \boldsymbol{x}_{-i})] \leq \lambda \cdot \mathbb{E}[C(\boldsymbol{x}^*)] + \mu \cdot \mathbb{E}[C(\boldsymbol{x})].$$

The "robust price of anarchy" (RPOA) is defined $\rho := \frac{\lambda}{1-\mu}$.

In fact, we can sample a CCE that approximately maximizes $\nu = -u_0$ with run-time polynomial in the game size, smoothness (λ_G, μ_G) of the original game G and the smoothness $(\lambda_{\tilde{G}}, \mu_{\tilde{G}})$ of a modified game $\tilde{G}$. Here, $\tilde{G}$ is identical to G except each agent's utility is changed from u_i to ν.

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

$$\overline{\nu}(\boldsymbol{x}) \geq \frac{y}{\rho} - \epsilon.$$

for any $y \leq \max_{\boldsymbol{x}^* \in \text{CCE}} \overline{\nu}(\boldsymbol{x}^*)$, where $\rho = \frac{\lambda_{\tilde{G}}}{1-\max\{\mu_G, \mu_{\tilde{G}}\}}$.

Proof Sketch of Theorem 1. First, we observe that the problem of finding a ν-maximizing CCE reduces to finding a halfspace oracle that optimizes some modified social welfare (Lemma 1). Similar reductions have been described by Jiang and Leyton-Brown (2011) and Barman and Ligett (2015).

Lemma 1. Fix a $0 < y \leq \max_{\boldsymbol{x} \in \text{CCE}} \overline{\nu}(\boldsymbol{x})$. Assume there is a Poly$(1/\epsilon, n, m)$-time halfspace oracle that, given a vector $\beta \in \mathbb{R}_+^{1+m}$ with non-negative components, returns an $\boldsymbol{x} \in P(A)$ such that $\beta v(\boldsymbol{x}) \leq 0$, where

$$v(\boldsymbol{x}) = \begin{bmatrix} \overline{u}_1(1, \boldsymbol{x}_{-1}) - \overline{u}_1(\boldsymbol{x}) \\ \vdots \\ \overline{u}_1(m_1, \boldsymbol{x}_{-1}) - \overline{u}_1(\boldsymbol{x}) \\ \vdots \\ \overline{u}_N(1, \boldsymbol{x}_{-N}) - \overline{u}_N(\boldsymbol{x}) \\ \vdots \\ \overline{u}_N(m_N, \boldsymbol{x}_{-N}) - \overline{u}_N(\boldsymbol{x}) \\ y - \overline{\nu}(\boldsymbol{x}) \end{bmatrix} \tag{3}$$

Then, there is a Poly$(1/\epsilon, n, m)$-time algorithm that returns an ϵ-CCE $\boldsymbol{x}$ with $\overline{\nu}(\boldsymbol{x}) \geq y - \epsilon$.

The halfspace oracle's optimization task can be reduced to optimizing social welfare in a game with RPOA of ρ, which inherits its smoothness from games G and $\tilde{G}$. This upper bounds the ratio between the smallest objective value ν of a CCE and the largest objective value ν of any strategy. Thus, we can use no-regret dynamics (Cesa-Bianchi and Lugosi 2006; Foster and Vohra 1997; Hart and Mas-Colell 2000) in this smooth game to approximate the half-space oracle.

Lemma 2. Let ρ denote an upper bound on the 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 $y \leq \rho \max_{\boldsymbol{x} \in \text{CCE}} \overline{\nu}(\boldsymbol{x})$, returns an $\boldsymbol{x} \in P(A)$ such that $\beta v(\boldsymbol{x}) \leq \epsilon$ where v is defined as in Equation 3.

Combining Lemmas 1 and 2 constructs our algorithm. □

Informally, this theorem states that it is tractable to find a CCE that maximizes ν up to the price of anarchy. While this relationship is immediate when ν 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 price-of-anarchy bounds and bounds on the tractability of CCE optimization.

Corollary 1. In linear congestion games, for any linear function ν, we can find, in Poly$(1/\epsilon, n, m)$ time, an ϵ-CCE $\boldsymbol{x}$ such that $\overline{\nu}(\boldsymbol{x}) \geq 0.4 \cdot \max_{\boldsymbol{x} \in \text{CCE}} \overline{\nu}(\boldsymbol{x}) - \epsilon$.

While Barman and Ligett (2015) showed the decision problem Γ of finding a non-trivial CCE is NP-hard in general, our Theorem 1 also shows it is tractable in smooth games.

Corollary 2. The decision-problem Γ is in P for games where $\frac{1}{\rho} \max_{\boldsymbol{x} \in \text{CCE}} \overline{\nu}(\boldsymbol{x}) > \min_{\boldsymbol{x} \in \text{CCE}} \overline{\nu}(\boldsymbol{x})$.

Remarks. These theoretical conclusions suggest that although using equilibria-based uncertainty sets may be intractable in some cases, there is a broad class of common problems where CCE uncertainty sets are reasonable and allow for efficient adversarial sampling. The algorithm we construct in Theorem 1 also enjoys two nice properties. First, it only requires oracle access to utility functions (efficient under polynomial expectation property). Second, in the algorithm's self-play subprocedures, each agent can be trained using only their, and their principal's, utility information.

Weakening Assumptions on Agents and Principal Robustness Algorithm

We now further refine our choice of uncertainty set to ensure generalization to agents that violate our behavioral assumptions. We now switch to weaker assumptions:

1. **Subjective rationality:** At test time, an agent's utility $\tilde{u}_i$ may differ from the anticipated utility u_i (Simon 1976). Many models of subjective rationality, such as Subjective Utility Quantal Response (Nguyen et al. 2013), bound the gap between u_i and $\tilde{u}_i$ as $\|u_i - \tilde{u}_i\|_\infty \leq \gamma_s$ with $\gamma_s > 0$.
2. **Procedural rationality:** An agent may not fully succeed in maximizing their utility (Simon 1976), e.g., they could gain up to $\gamma_p > 0$ utility if they unilaterally deviate.
3. **Myopia:** An agent may possess commitment power or otherwise be non-myopic, factoring in long-term incentives with a discount factor $\gamma_m \in (0, 1)$. This relates to notions of exogenous commitment power, e.g., partial reputation, in Stackelberg games (Kreps and Wilson 1982; Fudenberg and 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 even when presented with agents possessing these variations. Hence, we aim to use uncertainty sets $\boldsymbol{X}$ that encode such behaviors. The next proposition suggests it suffices to simply relax our uncertainty set $\boldsymbol{X}$ to include more approximate equilibria.

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

$$\text{CCE}_\varepsilon = \{\boldsymbol{x} \in P(A) \mid \forall i \in [1, \ldots, N], \tilde{a}_i \in A_i : \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{x}}[u_i(\boldsymbol{a})] + \varepsilon_i \geq \mathbb{E}_{\boldsymbol{a}_{-i} \sim \boldsymbol{x}_{-i}}[u_i(\tilde{a}_i, \boldsymbol{a}_{-i})]\},$$

where $\varepsilon_i = \max\{\frac{\|u_i\|_\infty}{1-\gamma_m}, \gamma_s, 2\gamma_p\}$. Hence, we can train policies robust to such agents by using the following in the robustness objective of Equation 1:

$$\boldsymbol{X}_\varepsilon(x_0) = \{\boldsymbol{x} \in P(A) \mid \exists x_0 \in A_0 : (x_0, \boldsymbol{x}) \in \text{CCE}_\varepsilon\}.$$

This proposition motivates us to use approximate CCE as an uncertainty set rather than exact CCEs. Conveniently, the algorithm we construct in the proof of Theorem 1 can be modified to adversarially sample from ϵ-CCE instead of exact CCE. By relaxation of Lemma 1, it yields the same optimality and runtime guarantees as Theorem 1, but over the set of approximate ϵ-CCE. We will refer to this modification of the Theorem 1 algorithm as Algorithm 2, which we repeat in full in the Appendix.

Finding Approximate Uncertainty Sets with Blackbox Optimizers

One challenge with using Algorithm 2 in practice is that it relies on a no-regret learning subprocedure that does not scale well. This is a common bottleneck in multi-agent learning when deriving practical algorithms from algorithms with strong theoretical guarantees. A common remedy is to replace the no-regret learning procedure with a standard learning algorithm (e.g., SGD), usually with no negative impact on empirical behavior or performance. We will do this to derive a practical variant of Algorithm 2 that still inherits its theoretical intuitions. This involves two main steps.

Removing binary search. Algorithm 2's binary search over y is a theoretically efficient search over possible optimal or worst-case values of ϵ-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 β_{m+1}, we can fix v_{m+1} to a sufficiently small value such that $\beta_{m+1} = 1[v_1, \dots, v_m \leq 0]$.

Replacing Blackwell's algorithm. During Algorithm 2's reduction to halfspace oracles, we can merge components of v corresponding to the same agent. This yields a functional equivalent of Eq 3,

$$v(x) = \begin{bmatrix} \max_{a_1 \in A_1} \overline{u}_1(a_1, \boldsymbol{x}_{-1}) - \overline{u}_1(\boldsymbol{x}) \\ \vdots \\ \max_{a_n \in A_n} \overline{u}_n(a_n, \boldsymbol{x}_{-n}) - \overline{u}_n(\boldsymbol{x}) \\ \sigma \to 0 \end{bmatrix}. \quad (4)$$

In practice, this choice is more tractable than Eq 3. Eq 4 lends itself to many efficient approximations. For instance, when A is combinatorially large, we can approximate the regret estimates with local methods rather than explicitly enumerating all possible action deviations.

A practical algorithm. Combined, these two modifications to Algorithm 2 render it equivalent to computing an upper-bound for the Lagrangian dual problem, $L(\epsilon)$, of adversarial sampling (Equation 6),

$$L(\epsilon) = \min_{\boldsymbol{x} \in P(A)} \max_{\lambda} \overline{u}_0(\boldsymbol{x}) - \sum_{i=1}^{N} \lambda_i \left[\text{Reg}_i(\boldsymbol{x}) - \epsilon \right], \quad (5)$$

$$\text{Reg}_i(\boldsymbol{x}) := \max_{a_i \in A_i} \overline{u}_i(a_i, \boldsymbol{x}_{-i}) - \overline{u}_i(\boldsymbol{x}).$$

Here, decoupled no-regret dynamics efficiently approximate the outer optimization $\min_{\boldsymbol{x} \in P(A)}$. In this sense, the theoretical results of Section can be interpreted as a formal bound on how much decoupled approximations of $\min_{\boldsymbol{x} \in P(A)}$ affect the lower-bound of this dual problem. Theorem 1 can thus be interpreted as the implication that, when playing a sufficiently smooth game, it is reasonable to use decoupled algorithms to approximate the dual problem. This motivates our final modification to Algorithm 2: replacing the inner no-regret learning loop with a blackbox self-play algorithm. The final algorithm is described in Algorithm 1.

Algorithm 1: Decoupled sampling of pessimistic equilibria.

Output: Approximate lower-bound on $L(\epsilon)$ (Eq 5).
Input: Number of training steps M_{tr} and self-play steps M_s, reward slack ϵ, multiplier learning rate α_λ, uncoupled self-play algorithm B, regret estimators $R_i : P(A) \to \mathbb{R}$ for each agent i.
Initialize mixed strategy $\boldsymbol{x}_1$.
for $j = 1, \dots, M_{\text{tr}}$ **do**
  for $i = 1, \dots, N$ **do**
    Estimate regret r_i as $\hat{r}_i \leftarrow R_i(\boldsymbol{x}_j)$, where $r_i := \max_{\tilde{x}_i \in P(A_i)} \overline{u}_i(\tilde{x}_i, \boldsymbol{x}_{-i}) - \overline{u}_i(\boldsymbol{x})$.
    Compute multiplier $\lambda_i \leftarrow \lambda_i - \alpha_\lambda (\hat{r}_i - \epsilon)$.
  end for
  Using B, run M_s rounds of self-play with utilities $\hat{u}_i(\boldsymbol{a}) := (\lambda_i u_i(\boldsymbol{a}) - u_0(\boldsymbol{a}))/(1 + \lambda_i)$.
  Set $\boldsymbol{x}_{j+1}$ as the resulting empirical play distribution.
end for
Return $\frac{1}{M_{\text{tr}}} \sum_{t=1}^{M_{\text{tr}}} \overline{u}_0(\boldsymbol{x}_t)$.

Adversarial Sampling Experiments

Before applying Algorithm 1 to more ambitious mechanism design tasks, we first benchmark the quality of its adversarial sampling. As our eventual mechanism design applications are spatiotemporal games requiring multi-agent reinforcement learning (MARL), for this experiment, we will also use MARL as the blackbox self-play procedure of Algorithm 1. In particular, we will use a common multi-agent implementation of the PPO algorithm (Schulman et al. 2017) and a Monte-Carlo sampling scheme as our regret estimator.

Game Environment. The game environment for this experiment is a Sequential Bimatrix Game. This is 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 4×4 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). Each episode is 500 timesteps.

We configure 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. The principal is a passive observer that observes the game and receives a payoff $r_0(s_i, s_j)$. The principal does not take any actions and its payoff is constructed

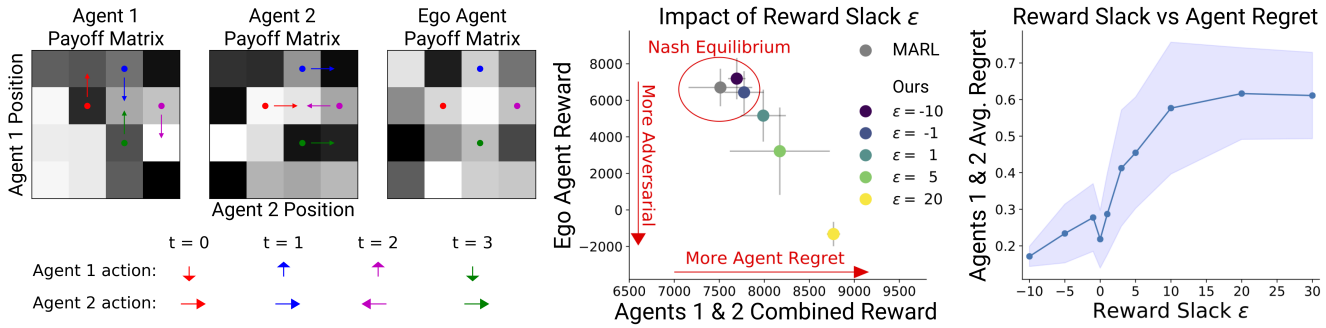

Figure 2: Validating our algorithm (Algorithm 3) in constrained repeated bimatrix games. Left: In the repeated bimatrix game, two agents navigate a 2D landscape. Both agents and the principal receive rewards based on visited coordinates. Brighter squares indicate higher payoff. The bimatrix reward structure encodes a social dilemma featuring a Nash equilibrium with low reward for the two agents and high reward for the gambler. Vanilla MARL converges to this equilibrium. Right: Agents trained with our algorithm deviate from this equilibrium in order to reduce the reward of the principal. ϵ governs the extent of the allowable deviation. As ϵ increases, the average per-timestep regret experienced by the agents also increases. Each average is taken over the final 12 episodes after rewards have converged. Each point in the above scatter plots describes the average outcome at the end of training for the agents (x-coordinate) and the principal (y-coordinate). Error bars indicate standard deviation.

such that its reward is high when the agents are at the Nash equilibrium. If Algorithm 1 successfully samples realistic worst-case behaviors, we expect to see agents 1 and 2 learning to deviate from their tragedy-of-the-commons equilibrium in order to (1) reduce the principal's reward but also (2) without significantly increasing their own regret.

Algorithm 1 efficiently interpolates between adversarial and low-regret equilibria. In Figure 2 (middle), we see the equilibria reached by agents balance the reward of the principal (y-axis) and themselves (x-axis). In particular, we see that conventional multi-agent RL discovers the attractive Nash equilibrium, which is in the top left. At this equilibrium, the agents do not cooperate and the principal receives high reward. Similarly, for small values of ϵ, our algorithm discovers the Nash equilibrium. Because ϵ constrains agent regret, with larger values of ϵ, our algorithm deviates farther from the Nash equilibrium, discovering ϵ-equilibria to the bottom-right that result in lower principal rewards.

Algorithm 1 has tight control over how much agents sacrifice to hurt the principal. We see in Figure 2 (right) that deviations from the Nash equilibrium yield higher regret for the agents, i.e., regret increases with ϵ. This figure also confirms that increasing our algorithm's slack parameter correctly increases the incentive of the agents to incur regret in order to harm the principal.

Optimizing Strategic Robustness

We now apply our adversarial sampling scheme, Algorithm 1, to automated mechanism design problems. In particular, we will use Algorithm 1 to provide feedback to a reinforcement learning (RL) procedure that selects mechanisms. This RL procedure, Algorithm 3, is described in the Appendix for completeness. First, we induce a mechanism design problem on a repeated n-matrix game. Then, we'll seek to learn an optimal tax policy in the AI Economist, a large-scale spatiotemporal simulation of an economy (Zheng et al. 2020). Each experiment setting features 4 to 5 agents involved in complex multi-timestep interactions. They are thus significantly more complex and costly to train in than traditional multi-agent RL environments.

Repeated Matrix Games

We first extend our repeated bimatrix game to include additional players and a principal.

Setup. The setting is now a 4-player, general-sum, normal-form game on a randomly generated $7\times7\times7\times7$ payoff matrix, with the same action and payoff rules as shown in Figure 2. Recall that these players will engage one another for 500 timesteps in an episode. Each player i is associated with an "original" reward function r_i; this is the reward function we will have access to during training. During test time, we may also encounter other categories of agents that have different reward functions but who are also themselves learned with RL: (1) *Vanilla*: r_i; (2) *Adversarial* (Adv): $r_i' = r_i - Qr_0$, where larger Q is more adversarial; and (3) *Risk-averse* (RiskAv): $r_i' = (r_i^{1-\eta} - 1)/(1-\eta)$, where higher η is more risk averse.

Results. Figure 1 shows the average rewards of principals trained (rows) and evaluated (columns) on each type of agents. We observe three key trends. First, principals perform better when evaluated on the same type of agent they were trained trained on (diagonal entries). Second, principals trained with our adversarial sampling perform better across the board. Third, the robustness gains of adversarial sampling are stronger when ϵ is large. This is expected as ϵ parameterizes the adversarial strength of our sampling scheme. For small ϵ, adversarial sampling reduces to random sampling as the CCE constraint is so tight it permits no adversarial deviations. We also saw that, even though all methods were run with 20 seeds and filtered down to 10 seeds on a validation set, our algorithm's results remain somewhat noisy, as it may not converge when badly initialized.

Training ↓ Testing →	**Original**	**Adv** ($Q = 0.25$)	**Adv** ($Q = 1$)	**RiskAv** ($\eta = 0.05$)	**RiskAv** ($\eta = 0.2$)
MARL	104±50.5	-5.8±34.1	-232±29.3	383±14.4	352±42.6
Adv ($Q = 0.25$)	**143**±35.0	⋆ **64.7**±35.1	-191±36.0	236±23.8	292±28.4
Adv ($Q = 1$)	131±63.1	-23±10.1	⋆ **-47**±8.20	286±35.1	290±36.0
RiskAv ($\eta = 0.05$)	-20±44.2	-112±15.7	-222±29.3	⋆ **404**±47.2	⋆ **464**±0.95
RiskAv ($\eta = 0.2$)	-53±32.4	-150±20.0	-283±29.8	⋆ **465**±0.61	**358**±70.9
Ours $\epsilon = -10$	⋆ **227**±50.8	⋆ **48.9**±12.4	**-137**±43.1	265±41.8	292±38.3
Ours $\epsilon = 50$	⋆ **221**±80.8	⋆ **48.3**±29.7	⋆ **-53**±14.0	⋆ **460**±33.9	⋆ **481**±38.6

Table 1: Robust performance in N-agent matrix games. We train an principal in a $7 \times 7 \times 7 \times 7$ matrix game with $n = 4$ agents (including the principal) 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 principal is trained on. 'MARL' refers to agents trained using their 'Original' reward definition; 'Adv' refers to adversarial agents; 'RiskAv' refers to risk-averse agents. The principals trained on these types of agents tend to perform best when evaluated on the same type seen during training. In contrast, principals 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**	$\eta = 0.11$	$\eta = 0.19$	$\eta = 0.27$	$\alpha = 0.25$	$\alpha = 2.5$
Free Market	326±1	527±2	427±1	**162**±1	248±2	112±0
Federal	335±8	637±5	497±2	150±2	⋆ **270**±1	121±0
Saez	⋆ **381**±1	597±3	487±4	⋆ **189**±1	265±0	127±0
Ours ($\epsilon = -30$)	⋆ **375**±9	**646**±6	**514**±12	**164**±9	**266**±2	**131**±2
AI Economist (Original)	⋆ **386**±2	628±5	**515**±1	123±13	**267**±0	129±1
AI Economist ($\eta = 0.11$)	253±5	⋆ **683**±7	506±1	140±1	255±2	129±0
AI Economist ($\eta = 0.19$)	308±17	**665**±9	⋆ **543**±6	82±29	256±3	**131**±1
AI Economist ($\eta = 0.27$)	339±11	603±3	477±1	137±10	**266**±0	129±0
AI Economist ($\alpha = 0.25$)	324±2	625±10	501±7	121±25	263±0	128±0
AI Economist ($\alpha = 2.5$)	104±27	636±3	246±33	49±10	251±4	⋆ **135**±1

Table 2: Robust dynamic tax policies in a spatiotemporal economy. First 3 rows are classic tax baselines: "Free Market" has no taxes; "US Federal" uses the 2018 US Federal progressive income tax rates; "Saez" uses an adaptive, theoretical formula to estimate optimal tax rates. Bottom rows correspond to learned policies trained to optimize, and evaluated on, the social welfare metric *swf* of equality and productivity. 'Ours' and 'AI Economist (Original)' are trained on the 'Original' settings (risk aversion $\eta = 0.23$; entropy bonus $\alpha = 0.025$). Naive multi-agent reinforcement learning tax policies, including (Zheng et al. 2020)'s original AI Economist, fail to generalize to previously unseen agent types. In contrast, our algorithm performs within standard error of top-1 on all agent types.

Taxing a Simulated Economy

We now apply our algorithm to designing dynamic (multi-timestep) tax policies for a simulated trade-and-barter economy with strategic taxpayers that interact with one another (Zheng et al. 2020). See Appendix for a screenshot.

Setup. In this simulated economy, the principal sets a tax policy and the agents play a partially observable game, given the tax policy. Each episode is 1,000 timesteps of economic activity. Taxpayers earn income $z_{i,t}$ from labor $l_{i,t}$ and pay taxes $T(z_{i,t})$. They optimize their expected isoelastic utility:

$$\tilde{z}_{i,t} = z_{i,t} - T(z_{i,t}), \quad r_{i,t}(\tilde{x}_{i,t}, l_{i,t}) = \frac{\tilde{x}_{i,t}^{1-\eta} - 1}{1-\eta} - l_{i,t},$$

where $\tilde{x}_{i,t}$ is the post-tax endowment of agent i, and $\eta > 0$ sets the degree of risk aversion (higher η means higher risk aversion) (Arrow 1971). Players expend labor and earn income by participating in a rich simulated grid-world with resources and markets. The principal optimizes for social welfare $swf = \left(1 - \frac{N}{N-1} gini(z)\right) \cdot \left(\sum_{i=1}^{N} z_i\right)$, the product of equality (Gini 1912) and productivity. The taxpayers are also themselves learned with multi-agent RL, using a PPO algorithm entropy hyperparameter α (Schulman et al. 2017).

Results. Figure 2 shows the social welfare achieved by our algorithm, naive dynamic RL policies (Zheng et al. 2020), and static baseline tax policies (Saez, US Federal). Naive RL policies achieve good test performance when evaluated on the same agents seen in training, but perform poorly with agents with different η and noise level. They are often outperformed by the baseline taxes, which perform surprisingly well under strong risk aversion ($\eta = 0.27$) and noisy agents (entropy bonus $\alpha = 0.25, 2.5$). We see that the static baseline taxes may be more robust than dynamic ones, even in complex environments. However, our algorithm closes this robustness gap, consistently outperforming or tying both AI Economists and baseline taxes.

Future Work

Efficient sampling of worst-case equilibria is a key challenge for robust decision-making, and by extension, automated mechanism design. As we've explored uncertainty sets based on game-theoretic concepts, future work may build uncertainty sets that use domain knowledge or historical data and that may yield robustness to other types of domain shifts, e.g., in game dynamics.

References

Aghassi, M.; and Bertsimas, D. 2006. Robust game theory. *Mathematical Programming*, 107(1): 231–273.

Arrow, K. J. 1971. The theory of risk aversion. *Essays in the theory of risk-bearing*, 90–120.

Barman, S.; and Ligett, K. 2015. Finding Any Nontrivial Coarse Correlated Equilibrium Is Hard. *ACM SIGecom Exchanges*, 14.

Basilico, N.; Coniglio, S.; Gatti, N.; and Marchesi, A. 2020. 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.

Bonabeau, E. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99: 7280–7287.

Cesa-Bianchi, N.; and Lugosi, G. 2006. *Prediction, Learning, and Games*. Cambridge University Press. ISBN 978-0-521-84108-5.

Conitzer, V.; and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce - EC '06*, 82–90. ACM Press. ISBN 978-1-59593-236-5.

Daskalakis, C.; Goldberg, P. W.; and Papadimitriou, C. H. 2009. The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing*, 39(1): 195–259.

Duetting, P.; Feng, Z.; Narasimhan, H.; Parkes, D. C.; and Ravindranath, S. S. 2019. Optimal Auctions through Deep Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 1706–1715. PMLR.

Foster, D. P.; and Vohra, R. V. 1997. Calibrated Learning and Correlated Equilibrium. *Games and Economic Behavior*, 21(1): 40–55.

Fudenberg, D.; and Levine, D. K. 1989. Reputation and Equilibrium Selection in Games with a Patient Player. *Econometrica*, 57(4): 759–778.

Gini, C. 1912. *Variabilità e mutabilità*. Tipogr. di P. Cuppini.

Gray, J.; Lerer, A.; Bakhtin, A.; and Brown, N. 2021. Human-Level Performance in No-Press Diplomacy via Equilibrium Search. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hart, S.; and Mas-Colell, A. 2000. A Simple Adaptive Procedure Leading to Correlated Equilibrium. *Econometrica*, 68(5): 1127–1150.

Hazan, E.; and Koren, T. 2016. The computational power of optimization in online learning. In Wichs, D.; and Mansour, Y., eds., *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 128–141. ACM.

Holland, J. H.; and Miller, J. H. 1991. Artificial Adaptive Agents in Economic Theory. *American Economic Review*, 81(2): 365–71.

Hou, L.; Pang, L.; Hong, X.; Lan, Y.; Ma, Z.; and Yin, D. 2020. Robust Reinforcement Learning with Wasserstein Constraint.

Jiang, A. X.; and Leyton-Brown, K. 2011. 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, 218–229. Springer. ISBN 978-3-642-25510-6.

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

Korzhyk, D.; Conitzer, V.; and Parr, R. 2011. Security Games with Multiple Attacker Resources. In Walsh, T., ed., *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, 273–279. IJCAI/AAAI.

Kreps, D. M.; and Wilson, R. 1982. Reputation and imperfect information. *Journal of Economic Theory*, 27(2): 253–279.

Li, S.; Wu, Y.; Cui, X.; Dong, H.; Fang, F.; and Russell, S. J. 2019. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 4213–4220. AAAI Press.

Littlestone, N. 1987. Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm. *Mach. Learn.*, 2(4): 285–318.

Morimoto, J.; and Doya, K. 2000. Robust Reinforcement Learning. In Leen, T. K.; Dietterich, T. G.; and Tresp, V., eds., *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, 1061–1067. MIT Press.

Myerson, R. B. 2016. Mechanism Design. In *The New Palgrave Dictionary of Economics*, 1–13. Palgrave Macmillan UK. ISBN 978-1-349-95121-5.

Nguyen, T. H.; Sinha, A.; Gholami, S.; Plumptre, A.; Joppa, L.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; and Critchlow, R. 2016. CAPTURE: A new predictive anti-poaching tool for wildlife protection. *Proceedings of 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Nguyen, T. H.; Yang, R.; Azaria, A.; Kraus, S.; and Tambe, M. 2013. Analyzing the Effectiveness of Adversary Modeling in Security Games. In desJardins, M.; and Littman, M. L., eds., *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*. AAAI Press.

Papadimitriou, C. H.; and Roughgarden, T. 2008. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3): 14:1–14:29.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust Adversarial Reinforcement Learning. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 2817–2826. PMLR.

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

Pita, J.; Jain, M.; Tambe, M.; Ordóñez, F.; and Kraus, S. 2010. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15): 1142–1171.

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

Roughgarden, T. 2015. Intrinsic Robustness of the Price of Anarchy. *Journal of the ACM*, 62(5): 32:1–32:42.

Schäfer, F.; Zheng, H.; and Anandkumar, A. 2020. Implicit competitive regularization in GANs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 8533–8544. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms.

Simon, H. A. 1976. 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*, 65–86. Springer US. ISBN 978-1-4613-4367-7.

Tessler, C.; Efroni, Y.; and Mannor, S. 2019. Action Robust Reinforcement Learning and Applications in Continuous Control. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 6215–6224. PMLR.

Zheng, S.; Trott, A.; Srinivasa, S.; Naik, N.; Gruesbeck, M.; Parkes, D. C.; and Socher, R. 2020. The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies.