ERMAS: LEARNING POLICIES ROBUST TO REALITY GAPS IN MULTI-AGENT SIMULATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Policies for real-world multi-agent problems, such as optimal taxation, can be learned in multi-agent simulations with AI agents that emulate humans. However, simulations can suffer from reality gaps as humans often act suboptimally or optimize for different objectives (i.e., bounded rationality). We introduce ϵ-Robust Multi-Agent Simulation (ERMAS), a robust optimization framework to learn AI policies that are robust to such multi-agent reality gaps. The objective of ERMAS theoretically guarantees robustness to the ϵ-Nash equilibria of other agents – that is, robustness to behavioral deviations with a regret of at most ϵ. ERMAS efficiently solves a first-order approximation of the robustness objective using meta-learning methods. We show that ERMAS yields robust policies for repeated bimatrix games and optimal adaptive taxation in economic simulations, even when baseline notions of robustness are uninformative or intractable. In particular, we show ERMAS can learn tax policies that are robust to changes in agent risk aversion, improving policy objectives (social welfare) by up to 15% in complex spatiotemporal simulations using the AI Economist (Zheng et al., 2020).

1 INTRODUCTION

Reinforcement learning (RL) offers a tool to optimize policy decisions affecting complex, multi-agent systems; for example, to improve traffic flow or economic productivity. In practice, the need for efficient policy evaluation necessitates training on simulations of multi-agent systems (MAS). Agents in these systems can be emulated with fixed behavioral rules, or by optimizing for a reward function using RL (Zheng et al., 2020). For instance, the impact of economic policy decisions are often estimated with agent-based models (Holland & Miller, 1991; Bonabeau, 2002). This commonly introduces a *reality gap* as the reward function and resulting behavior of simulated agents might differ from those of real people (Simon & Schaeffer, 1990). This becomes especially problematic as the complexity of the simulation grows, for example, when increasing the number of agents, or adding agent affordances (Kirman, 1992; Howitt, 2012). As a result, policies learned in imperfect simulations need to be robust against reality gaps in order to be effective in the real world.

We introduce ϵ-Robust Multi-Agent Simulation (ERMAS), a robust optimization framework for training robust policies, termed *planners*, that interact with real-world multi-agent systems. ERMAS trains robust planners by simulating multi-agent systems with RL and sampling *worst-case* behaviors from the *worst-case* agents. This form of multi-agent robustness poses a very challenging multi-level (e.g., max-min-min) optimization problem. Existing techniques which could be applied to ERMAS's multi-agent robustness objective, e.g., naive adversarial robustness (Pinto et al., 2017) and domain randomization (Tobin et al., 2017; Peng et al., 2018), are intractable as they would require an expensive search through a large space of agent reward functions. Alternative frameworks improve robustness, e.g., to changes in environment dynamics, observation or action spaces (Pinto et al., 2017; Li et al., 2019; Tessler et al., 2019), but do not address reality gaps due to reward function mismatches, as they use inappropriate metrics on the space of adversarial perturbations.

To solve this problem, ERMAS has three key features: 1) It formulates a multi-agent robustness objective equivalent to finding the worst case ϵ-Nash equilibria. 2) It optimizes a tractable dual problem to the equivalent objective. 3) It approximates the dual problem using local solution concepts and first-order meta-learning techniques (Nichol et al., 2018; Finn et al., 2017). ERMAS ultimately yields policies that are robust to other agents' behavioral deviations, up to a regret of ϵ.

We show that ERMAS learns robust policies in repeated bimatrix games by finding the worst-case reality gaps, corresponding to highly adversarial agents, which in turn leads to more robust planners. We further consider a challenging, large-scale spatiotemporal economy that features a social planner that learns to adjust agent rewards. In both settings, we show policies trained by ERMAS are more robust by testing them in perturbed environments with agents that have optimized for reward functions unused during ERMAS training. This generalization error emulates the challenge faced in transferring policies to the real world. In particular, we show ERMAS can find AI Economist tax policies that achieve higher social welfare across a broad range of agent risk aversion objectives. In all, we demonstrate ERMAS is effective even in settings where baselines fail or become intractable.

Contributions To summarize, our contributions are:

- We derive a multi-agent adversarial robustness problem using ϵ-Nash equilibria, which poses a challenging nested optimization problem.
- We describe how ERMAS efficiently solves the nested problem using dualization, trust-regions, and first-order meta-learning techniques.
- We empirically validate ERMAS by training robust policies in two multi-agent problems: sequential bimatrix games and economic simulations. In particular, ERMAS scales to complex spatiotemporal multi-agent simulations.

2 Robustness and Reality Gaps in Multi-agent Environments

Informally, we seek to learn a policy π_p, which we term the *planner*, that interacts with an environment featuring N other agents. The planner's objective depends both on its own policy and the behavior of the agents in response to that policy, leading to a multi-agent RL problem in which the planner and agents co-adapt. In practice, evaluating (and optimizing) π_p requires use of a simulation with agents that emulate those in the environment of interest (i.e. the real world), which might contain agents whose reward function differs from those used in the simulation. Our goal is to train planner policies that are robust to such reality gaps.

Formally, we build on partially-observable multi-agent Markov Games (MGs) (Sutton & Barto, 2018), defined by the tuple $M := (S, A, r, \mathcal{T}, \gamma, o, \mathcal{I})$, where S and A are the state and action spaces, respectively, and $\mathcal{I}$ are agent indices. Since the MG played by the agents depends on the choice of planner policy, we denote the MG given by π_p as $M[\pi_p]$. MGs proceed in episodes that last $H + 1$ steps (possibly infinite), covering H transitions. At each time $t \in [0, H]$, the world state is denoted s_t. Each agent $i = 1, \ldots, N$ receives an observation $o_{i,t}$, executes an action $a_{i,t}$ and receives a reward $r_{i,t}$. The environment transitions to the next state s_{t+1}, according to the transition distribution $\mathcal{T}(s_{t+1}|s_t, \boldsymbol{a}_t)$.[1] Each agent observes $o_{i,t}$, a part of the state s_t. Agent policies π_i are parameterized by θ_i while the planner policy π_p is parameterized by θ_p.

The Nash equilibria of $M[\pi_p]$ are agent policies where any unilateral deviation is suboptimal:

$$\texttt{ANE}(\pi_p) := \{\boldsymbol{\pi} \mid \forall i \in [1, N], \tilde{\pi}_i \in \Pi : J_i(\tilde{\pi}_i, \pi_{-i}, \pi_p) \leq J_i(\pi_i, \pi_{-i}, \pi_p)\}, \tag{1}$$

where $J_i(\boldsymbol{\pi}, \pi_p) := \mathbb{E}_{\boldsymbol{\pi}, \pi_p}\left[\sum_{t=0}^{H} \gamma^t r_t^{(i)}\right]$ denotes the objective of agent i. Hence, a rational agent would not unilaterally deviate from $\boldsymbol{\pi} \in \texttt{ANE}(\pi_p)$.

To evaluate a fixed planner policy π_p, we simply sample outcomes using policies $\boldsymbol{\pi} \in \texttt{ANE}(\pi_p)$. Also optimizing π_p introduces a form of *two-level learning*. Under appropriate conditions, this can be solved with simultaneous gradient descent (Zheng et al., 2020; Fiez et al., 2019).

Robustness Objective As noted before, we wish to learn planner policies π_p that are robust to reality gaps arising from changes in agent reward functions, e.g., when agents are boundedly rational.[2] We develop a robustness objective for the planner by formalizing such reality gaps as perturbations

[1] Bold-faced quantities denote vectors or sets, e.g., $\boldsymbol{a} = (a_1, \ldots, a_N)$, the action profile for N agents.

[2] This type of reality gap occurs when the simulated environment's reward function r fails to rationalize the actual behavior of the agents in the real environment, i.e., when agents in the real world act suboptimally with respect to the simulation's reward function.

$\xi_i \in \Xi$ to agent objectives, where the *uncertainty set* $\Xi : (S, A)^H \to \mathbb{R}$ is the space of possible perturbations. We extend $\texttt{ANE}(\pi_p, \boldsymbol{\xi})$ to condition the Nash equilibria on perturbations $\boldsymbol{\xi}$:

$$\texttt{ANE}(\pi_p, \boldsymbol{\xi}) := \{\boldsymbol{\pi} \mid \forall i \in [1, N], \tilde{\pi}_i \in \Pi : J_i^{\xi}(\tilde{\pi}_i, \pi_{-i}, \pi_p) \leq J_i^{\xi}(\pi_i, \pi_{-i}, \pi_p)\}, \tag{2}$$

$$J_i^{\xi}(\tilde{\pi}_i, \pi_{-i}, \pi_p) := J_i(\tilde{\pi}_i, \pi_{-i}, \pi_p) + \mathbb{E}_{\tau_i \sim \tilde{\pi}_i, \pi_{-i}, \pi_p}\left[\xi_i(\tau_i)\right] \tag{3}$$

where τ_i is a trajectory (sequence of state-action pairs). Following Morimoto & Doya (2001), a robust planner optimizes its reward, subject to agents playing a perturbed Nash equilibrium $\texttt{ANE}(\pi_p, \boldsymbol{\xi})$ that maximally penalizes the planner:

$$\pi_p^* = \arg\max_{\pi_p} \min_{\boldsymbol{\xi} \in \boldsymbol{\Xi}} \min_{\boldsymbol{\pi} \in \texttt{ANE}(\pi_p, \boldsymbol{\xi})} J_p(\boldsymbol{\pi}, \pi_p). \tag{4}$$

Note that agent policies $\boldsymbol{\pi} \in \texttt{ANE}(\pi_p, \boldsymbol{\xi})$ describes agents that optimize their own reward function, and we assume an adversary chooses $\boldsymbol{\xi}$.

Bounded Uncertainty Set There are two challenges with Equation 4. First, if the adversary can arbitrarily set $\boldsymbol{\Xi}$, the worst case is uninformative.[3] Second, depending on the complexity of Π, the uncertainty set $\boldsymbol{\Xi}$ may be high-dimensional and intractable to search.We address these issues by upper-bounding the L_∞ norm of $\xi_i \in \Xi$:

$$\Xi_\epsilon = \left\{ \xi \;\middle|\; \sup_{\boldsymbol{\pi}, \pi_p} |\xi_i(\boldsymbol{\pi}, \pi_p)| < \epsilon, \text{ for all } i \in I \right\}. \tag{5}$$

This uncertainty set is equivalent to the ϵ-equilibria of $M[\pi_p]$:

$$\texttt{ANE}(\pi_p, \epsilon) := \{\boldsymbol{\pi} \mid \forall i \in [1, N], \tilde{\pi}_i \in \Pi : J_i(\tilde{\pi}_i, \pi_{-i}, \pi_p) \leq J_i(\pi_i, \pi_{-i}, \pi_p) + \epsilon\}. \tag{6}$$

Intuitively, ϵ parameterizes the size of the reality gap to which the planner is robust. With this, the robustness objective over a bounded uncertainty set is:

$$\underbrace{\arg\max_{\pi_p} J^*_{p,\min}(\pi_p, \epsilon)}_{\texttt{Planner-OPT}}, \quad \text{where } J^*_{p,\min}(\pi_p, \epsilon) := \underbrace{\min_{\boldsymbol{\pi} \in \texttt{ANE}(\pi_p, \epsilon)} J_p(\boldsymbol{\pi}, \pi_p)}_{\texttt{Agent-Adv-Search}}. \tag{7}$$

Using $\texttt{ANE}(\pi_p, \epsilon)$ replaces the problem of intractably searching through $\boldsymbol{\Xi}$ with searching through $\texttt{ANE}(\pi_p, \epsilon)$, and thus merges the two nested $\min$ operations in Equation 4. Conceptually, this transfers the worst-case search problem to the agents: `Agent-Adv-Search` agents find an adversarial equilibrium; `Planner-OPT` optimizes the planner given adversarial agents.

3 ERMAS: Robust Policies in Multi-Agent Simulations

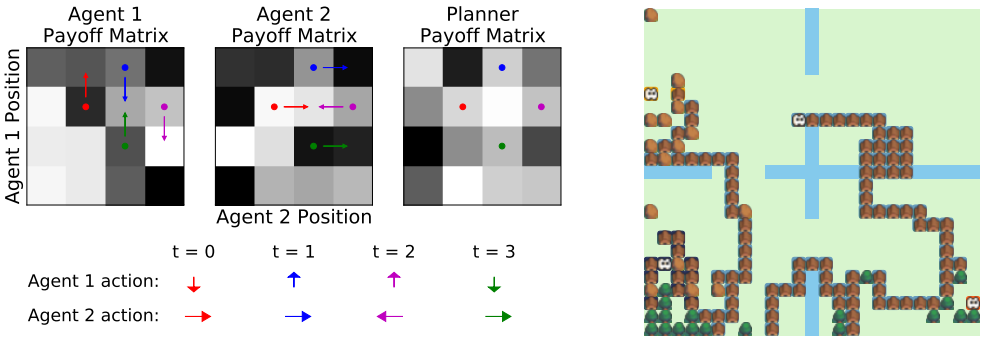
We now introduce ERMAS, an efficient optimization framework to solve the robustness problem in Equation 7. ERMAS proceeds in three steps. First, it dualizes Equation 7 following constrained RL. Second, it defines a trust-region for the uncertainty set $\texttt{ANE}(\pi_p, \epsilon)$, approximating the dual problem. Finally, it uses first-order meta-learning with trust-regions to solve the approximate dual problem. **See Appendix A.1 for the detailed Algorithm description.**

Dualizing `Agent-Adv-Search` The agent search problem in Equation 7 can be formulated similar to a constrained RL problem (Paternain et al., 2019), where the primary objective of the agents is to minimize the planner's reward and the secondary objective is to maximize their own reward. While conventional constrained RL enforces a constant lower bound in the constraint, e.g., $J_i(\boldsymbol{\pi}, \pi_p) \geq C$, we enforce a dynamic one: $\forall i \in 1 \ldots N : J_i(\boldsymbol{\pi}, \pi_p) \geq J_i(\pi_i^*, \pi_{-i}, \pi_p) - \epsilon$, where π_i^* is the optimal unilateral deviation for agent i: $\pi_i^* := \arg\max_{\tilde{\pi}_i \in \Pi} J_i(\tilde{\pi}_i, \pi_{-i}, \pi_p)$.

Specifically, we dualize `Agent-Adv-Search`, i.e., Equation 4, as:

$$\min_{\boldsymbol{\pi}} \underbrace{\left(J_p(\boldsymbol{\pi}, \pi_p) - \sum_{i=1}^{N} \lambda_i \left[J_i(\pi_i^*, \pi_{-i}, \pi_p) - J_i(\boldsymbol{\pi}, \pi_p) - \epsilon_i \right] \right)}_{J^{\dagger}_{p,\min}(\pi_p, \epsilon)}, \tag{8}$$

[3]For instance, by setting ξ_i such that $J_i^{\xi} = -J_p$.



(a) Repeated Bimatrix Game

(b) Spatiotemporal Economic Simulation

Figure 1: ERMAS trains planners that are robust to reality gaps in these testbed multi-agent simulations. **(Left)** In the repeated bimatrix game, a pair of agents navigates a 2D landscape. Both agents and the planner receive rewards based on visited coordinates. Brighter squares indicate higher payoff. **(Right)** In the spatiotemporal economic simulation, 4 heterogenous agents perform labor, trade resources, earn income, and pay taxes according to the schedules set by the planner.

where $\boldsymbol{\lambda}$ are Lagrange multipliers. We abuse notation to denote $\boldsymbol{\theta} := [\theta_1, \ldots, \theta_N, \theta_p]$ and $J_p(\boldsymbol{\theta}) := J_p(\boldsymbol{\pi}, \pi_p)$ where θ are the parameters of π. To solve Equation 8, the agents apply gradients:

$$\nabla_{\theta_i} J^{\dagger}_{p,\min}(\pi_p, \epsilon) = -\nabla_{\theta_i} J_p(\boldsymbol{\theta}) - \lambda_i \nabla_{\theta_i} \left[J_i(\theta'_i(\boldsymbol{\theta}), \theta_{-i}) - J_i(\boldsymbol{\theta}) \right], \tag{9}$$

where $\theta'_i(\boldsymbol{\theta})$ is the parameters of the optimal unilateral deviation π^*_i for agent i, i.e. the parameters that minimize local regret, which depends on the current policy parameters θ_i. λ_i is updated as:

$$\nabla_{\lambda_i} J^{\dagger}_{p,\min}(\pi_p, \epsilon) = J_i(\pi^*_i, \pi_{-i}, \pi_p) - J_i(\boldsymbol{\pi}, \pi_p) - \epsilon_i. \tag{10}$$

Equation 8 still poses a challenge through the $J_i(\pi^*_i, \pi_{-i}, \pi_p)$ terms, which correspond to unknown agent regret. We now detail the efficient approximation of the value and derivative of agent regret using local and meta-learning approximations, respectively.

Trust Regions using Local ϵ-equilibria Estimating regret requires knowledge of the optimal unilateral deviation for agent i. We can simplify this problem by proposing a refinement of ϵ-equilibria inspired by the notion of local Nash equilibria in differentiable games (Ratliff et al., 2014).

Definition 3.1. A strategy $\boldsymbol{\pi}$ is a local ϵ-Nash equilibrium if there exists open sets $W_i \subset \Pi^N$ such that $\pi_i \in W_i$ and for each $i \in \{1, \ldots, N\}$ we have that $J_i(\pi'_i, \pi_{-i}) \leq J_i(\boldsymbol{\pi}) + \epsilon'$ for all $\pi'_i \in W_i \setminus \{\pi_i\}$, where $\epsilon' := \epsilon \sup_{\pi'_i \in W_i} KL(\pi_i || \pi'_i)$.

By instead performing `Agent-Adv-Search` on the local ϵ-Nash equilibria, we can limit the set of unilateral deviations to consider to a small trust region, $\Pi_\eta(\pi)$:

$$\texttt{ANE}(\pi_p, \eta) := \{\boldsymbol{\pi} \mid \forall i \in [1, N], \tilde{\pi}_i \in \Pi_\eta(\pi_i) : J_i(\tilde{\pi}_i, \pi_{-i}, \pi_p) \leq J_i(\pi_i, \pi_{-i}, \pi_p) + \epsilon\}, \tag{11}$$

$$\Pi_\eta(\pi) := \{\pi' \in \Pi \mid KL(\pi || \pi') \leq \eta\}, \tag{12}$$

where $\eta > 0$ defines the size of the trust region. For small η, algorithms such as TRPO (Schulman et al., 2017) can be used to efficiently approximate optimal local deviations of π_i, affording reasonable approximations of $J_i(\pi^*_i, \pi_{-i}, \pi_p)$.

First-Order Meta Learning Approximation The full gradient in Equation 9 is also complicated by the need to estimate the derivative of local regret $\nabla_{\theta_i} \left[J_i(\theta'_i(\boldsymbol{\theta}), \theta_{-i}) - J_i(\boldsymbol{\theta}) \right]$. The second term maximizes the performance of the agent's policy and is simply found with policy gradient. The first term is less straightforward: it minimizes the performance of the best agent policy in the current trust region. We note that this first term corresponds to a meta-learning gradient. We follow REPTILE (Nichol et al., 2018) to obtain a first-order approximation of a M-step meta-learning gradient:

$$\nabla_{\theta_i} J_i(\theta'_i(\boldsymbol{\theta}), \theta_{-i}) = g_1 - \frac{1}{M} \sum_{i=1}^{M} g_i, \quad g_i = \nabla_{\theta_i} J_i \left(\theta_i + \sum_{j=1}^{i-1} g_j, \theta_{-i}, \theta_p \right), \tag{13}$$

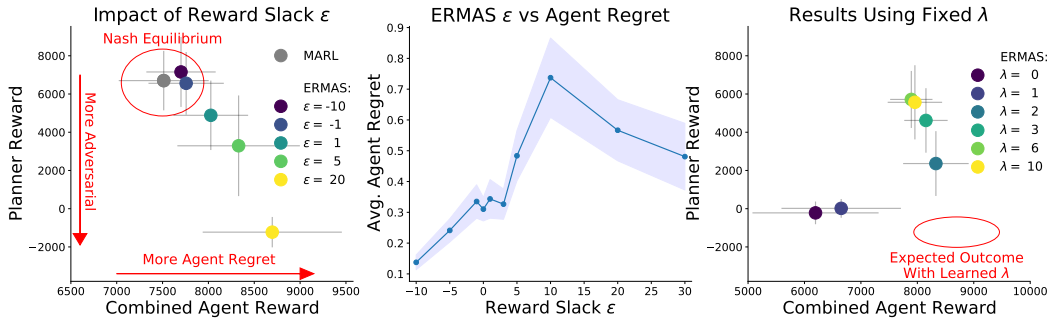


Figure 2: Validating ERMAS `Agent-Adv-Search` in the repeated bimatrix game. In this setting, the planner policy is fixed to emphasize the behavior of `Agent-Adv-Search`. Each point in the above scatter plots describes the average outcome at the end of training for the agents (x-coordinate) and the planner (y-coordinate). Error bars indicate standard deviation. **(Left)** The bimatrix reward structure encodes a social dilemma featuring a low-agent-reward/high-planner-reward Nash equilibrium, which is where vanilla MARL converges. Agents trained with ERMAS deviate from this equilibrium in order to reduce planner reward. ϵ governs the extent of the allowable deviation. **(Middle)** As ϵ increases, the average per-timestep regret experienced by the agents also increases. **(Right)** Using fixed values of λ (rather than allowing it to update, as in the full algorithm) distorts performance and prevents agents from reaching the same ϵ-equilibria discovered with learned λ.

where g_i denotes the ith policy gradient in the direction of J_i. In practice, we scale this meta-learning term with the hyperparameter β as $\beta < 1$ incorporates a helpful inductive bias where maximizing agent reward leads to local maxima. We can alternatively apply this gradient update periodically, to both mimic $\beta < 1$ and reduce computation overhead.

ERMAS By solving the dual problem (Eq. 8), ERMAS yields robustness to ϵ-equilibria and, equivalently, uncertainty in agent objectives. ERMAS solves the dual problem by combining trust regions and meta-learning techniques to estimate and differentiate agent regret. Algorithm 1 and 2 (Appendix A.1) describe an efficient implementation of this procedure for nested policy learning.

4 Experimental Validation in Multi-Agent Simulations

4.1 ERMAS Solves the Inner Loop: Agent-Adv-Search

Constrained Repeated Bimatrix Game We analyze the agent behaviors learned by `Agent-Adv-Search` in a modified repeated bimatrix game (Figure 1a). At each timestep t a row and a column player simultaneously choose from a finite number of actions. Each pair of actions (i, j) corresponds to a cell in the agent payoff matrices $r_1(i, j), r_2(i, j)$. At the initial timestep $t = 0$, the agents play a random pair of actions, corresponding to a random cell in the matrix. At each subsequent timestep $t > 0$, the agents can only choose an action adjacent to their previously selected action. This can be interpreted as "movement" through the payoff matrix with the row player (Agent 1) controlling up-down movement and the column player (Agent 2) controlling left-right movement. The planner observes the game and receives a payoff $r_p(i, j)$.

Discovering ϵ-Equilibria Due to its simplicity, we can explicitly detail the social dilemmas and game-theoretic dynamics in this game. Figure 2 (left) shows that agents using independent RL discover the Nash equilibrium located in the top left (high planner reward, low agent reward). There also exists a space of ϵ-equilibria characterized by high agent reward $\boldsymbol{r}$, low agent regret $J_i(\pi_i^*, \pi_{-i}) - J_i(\boldsymbol{\pi})$, and low planner reward $J_p(\boldsymbol{\pi})$. As ϵ increases, the agents' objective provides more "slack" for the agents to deviate from the stable, Nash equilibrium in order to penalize the planner. In essence, and as described by Equation 7, this adversarial behavior is key to the goal of learning a more robust planner. However, these ϵ-equilibria are inherently unstable, in this case due to the social dilemmas embedded in the reward structure. Agent regret, which tends to increase with ϵ (Figure 2, Middle), is a signature of this instability.

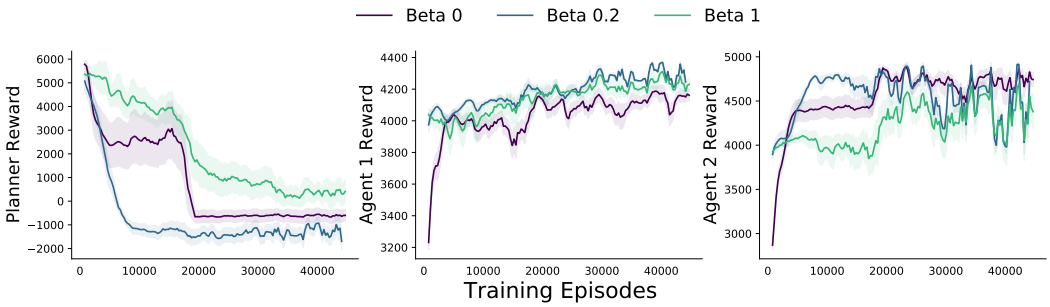


Figure 3: Planner and agent rewards in the repeated bimatrix game over training time. Each line represents an average over 10 seeds, with error bars indicating standard error. The lines correspond to runs of ERMAS and are colored according to β, the weight of the meta-learning term of Equation 13. **(Left)** Planner performance over training episodes. **(Middle)** Agent 1 performance over training episodes. **(Right)** Agent 2 performance over training episodes.

4.2 All the Components of ERMAS are Necessary

Dynamic or Frozen Lagrange multipliers λ_i The Lagrange multipliers $\boldsymbol{\lambda}$ balance the dual objectives of seeking stable agent equilibria and minimizing the planner's welfare. The $\boldsymbol{\lambda}$ are updated using local estimates of agent regret, as described in Equation 8. Are these updates necessary? Figure 2 (right) addresses this question. It shows the equilibria learned by ERMAS across varying values of frozen $\boldsymbol{\lambda}$. Freezing $\boldsymbol{\lambda}$ reduces Equation 8 to agents learning while sharing rewards. This can dramatically affect the equilibria discovered by ERMAS. Larger values of λ mean agents antagonize the planner less and improve agent and planner utility. However, the bottom right quadrant and its ϵ-equilibrium are never reached. In fact, certain ϵ-equilibria are only reachable with dynamic λ. Hence, with frozen $\boldsymbol{\lambda}$ the agents would be less effective in robustifying the planner.[4]

First-order Approximation Using the first order approximation of Equation 13 might slow learning, since we step "away" from the direction that the standard policy gradient suggests. In contrast, we find that the ERMAS update can significantly speed up convergence. Figure 3 depicts the learning curve of variants of ERMAS with varying weights, β, on the meta-learning term. For instance, when $\beta = 0$, the meta-learning term is completely dropped. We find that lines corresponding to larger values of β reach improved agent rewards half an order of magnitude faster than lower values of β. However, larger values of β fail to reach the desirable region of adversarial ϵ-equilibria. In both cases, very large ($\beta > \frac{1}{N}$) and very small ($\beta \sim 0$) fail to fully reach and remain at the adversarial region. However, an appropriate range for β gives rapid convergence and good asymptotic behavior.

4.3 Solving Planner-OPT Using Agent-Adv-Search

We now show that ERMAS yields planners π_p which are more robust to uncertainty in the agent's reward function. Specifically, we empirically validate that ERMAS can find strong solutions to the nested optimization problems `Agent-Adv-Search` and `Planner-OPT`, in two simulations: repeated bimatrix games and economic simulations (see Figure 1).

Evaluation Workflow To measure planner robustness, we evaluate trained planners in test environments with a reality gap (i.e. reward parameterizations that differ from the training environment), containing test agents that are optimized for the test environment's reward functions and that have not been seen by the planner. We proceed as follows: 1) Train the agents and planner with ERMAS. 2) Fix the planner and transfer it to a test environment, then train new agents *from scratch* in the test environment with reward functions unseen by the planner. 3) Report the value of the planner objective after the new agents have converged.

[4]However, we recommend temporarily freezing λ at the top of Algorithm 2 to "warm up" agent policies.

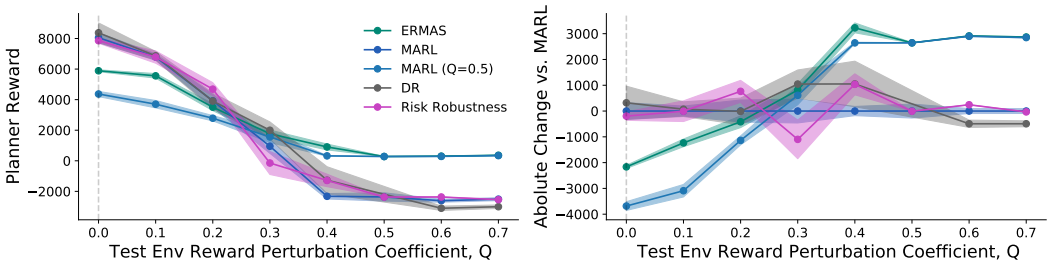


Figure 4: Robust planner learning in the repeated bimatrix game. Average planner performance as a function of the reward perturbation coefficient Q in the test environment. Each point represents an average over 5 seeds, with error bars indicating standard error. Vertical dashed lines denote the training environment Q. **(Left)** Average planner performance at convergence. **(Right)** Performance relative to the vanilla MARL baseline, which is trained without a robustness objective.

Augmented Repeated Bimatrix Games We modify the bimatrix game to let the planner observe the game state, set an integer $a \geq 0$, and get reward $ar_p(i,j) - a^2$. a scales the reward but adds a cost a^2, which counters greed (always choosing large a). Since the agents control i, j, they can (intentionally) reduce the planner's objective. During evaluation, the agents receive $r_i' = r_i - Qr_p$ where $Q \leq 1$ is a fixed parameter that models a reality gap. $Q = 0$ means no train/test reality gap.

Figure 4 shows the test performance of ERMAS against vanilla multi-agent RL policies learned without a reality gap ($Q_{\text{train}} = 0$), and a pessimistic setting $Q_{\text{train}} = 0.5$. Naturally, MARL $Q_{\text{train}} = 0.5$ underperforms $Q_{\text{train}} = 0$ at $Q_{\text{test}} < 0.3$, but keeps consistent performance throughout. `Agent-Adv-Search` successfully finds the worst-case perturbation, and yields a planner policy at least as good as MARL $Q_{\text{train}} = 0.5$. This shows that ERMAS can produce policies which are robust to multi-agent uncertainty and is a tractable adversarial robustness algorithm.

The baselines, domain randomization (DR) and risk robustness (RR), do not improve over vanilla robust multi-agent RL. For DR, agents receive $r_i' = r_i + \sigma$, where $\sigma \sim U[-1, 1]$ (recall $r_i \in [0, 1]$). Even in a simple 4-by-4 bimatrix game, there are $2304 = 16$ states $\times 16$ states $\times 9$ actions possible reward values to perturb; a 2304-dimensional space is too sparse to cover uniformly. Furthermore, in contrast to DR of visual inputs or dynamics, there is a natural latency to the effect of DR of agent rewards: randomization only has an effect if agents learn to adapt to their perturbed rewards. RR makes the planner's reward more concave, e.g., $\log r_p$. However, RR does not yield the multi-agent robustness relevant to our problem.

4.4 ERMAS FOR LARGE-SCALE TWO-LEVEL LEARNING: AI ECONOMIST

We train a robust AI Economist: a planner for optimal taxation that acts in a spatiotemporal economic simulation, see Zheng et al. (2020) for a detailed description. This setting is very challenging given the scale of the simulation and agent affordances. In particular, *adversarial and randomization baselines do not scale to this setting.* In this simulation, agents optimize their expected utility $\mathbb{E}_{\boldsymbol{\pi},\pi_p}\left[\sum_{t=0}^{T} r_{i,t}\right]$, where the utility $r_{i,t}$ is a function of labor $l_{i,t}$ and post-tax endowments $\tilde{x}_{i,t}$:

$$\tilde{z}_{i,t} = z_{i,t} - T(z_{i,t}), \quad \tilde{x}_{i,t} = \sum_{t' \leq t} \tilde{z}_{i,t}, \quad r_{i,t}(\tilde{x}_{i,t}, l_{i,t}) = \frac{\tilde{x}_{i,t}^{1-\eta} - 1}{1-\eta} - l_{i,t}, \quad \eta > 0, \tag{14}$$

where η sets the degree of risk aversion (higher η means higher risk aversion). Agents earn income $z_{i,t}$ and pay taxes $T(z_{i,t})$, which are set by the planner. The planner optimizes social welfare `swf`:

$$\texttt{swf} = \texttt{eq}(x) \cdot \texttt{prod}(x), \quad \texttt{eq}(x) = 1 - \frac{N}{N-1}\texttt{gini}(x), \quad \texttt{prod}(x) = \sum_{i=1}^{N} x_i, \tag{15}$$

a combination of equality (Gini, 1912) and productivity.

Figure 5 shows the performance of ERMAS tax policies along with baseline tax policies (Saez, US Federal) and an AI Economist (MARL). All models are trained on agents with $\eta = 0.23$. We

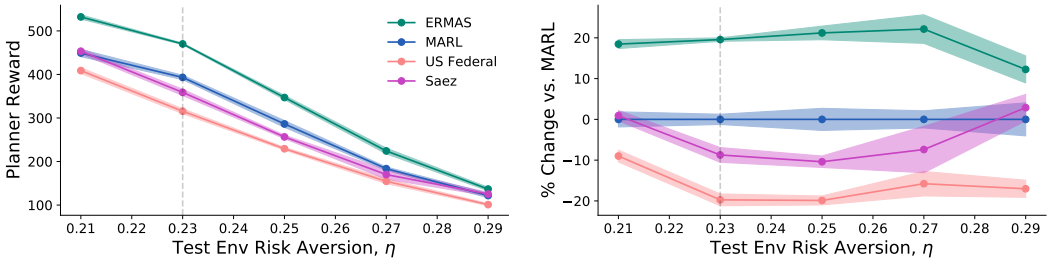


Figure 5: Learning robust optimal taxation in an economic simulation. Results are plotted following the same conventions in Figure 4. Due to the intractability of other robustness algorithms in this setting, we instead compare against different planner choices. "US Federal" uses a fixed tax scheme adapted from the 2018 US Federal income tax rates. "Saez" uses an adaptive, theoretical formula to estimate optimal tax rates.

replicate Zheng et al. (2020): MARL outperforms the Saez tax. However, MARL fails to outperform Saez when testing with $\eta < 0.22$ and $\eta > 0.28$. In contrast, ERMAS yields consistent gains across the depicted range of η. This shows that ERMAS's tax policy can improve `swf` even if agent risk aversion significantly increases or decreases (e.g., due to economic cycles or exogenous shocks).

In fact, ERMAS outperforms the baseline AI Economist for the original setting of $\eta = 0.23$. This suggests the robustness and performance do not necessarily pose a zero-sum game: ERMAS can find equilibria with high performance and strong generalization. Single-agent robust reinforcement learning algorithms similarly yield performance improvements even in the absence of environment perturbations—robust learning simply yields better policies Pinto et al. (2017).

5 Related Work

Sim2real: Reality Gaps The Sim2Real problem considers how reality gaps, i.e., mismatches between a simulation and reality, can (negatively) impact RL policies. Robust performance across reality gaps has been studied when applying deep RL to robotic control with visual inputs (James et al., 2019; Christiano et al., 2016; Rusu et al., 2016; Tzeng et al., 2015). To close reality gaps, domain adaptation aims to transfer a simulated distribution (of the world) to the real world (Higgins et al., 2017; Kim et al., 2019). In comparison, the study of reality gaps in multi-agent RL is relatively nascent (Suh et al., 2019; Nachum et al., 2019).

Multi-Agent Robustness Morimoto & Doya (2001) proposed robust RL for the worst-case adversarial objective (i.e., a max-min optimization problem), inspired by $\mathcal{H}_\infty$ control, and provided analytical solutions in the linear setting. The robust RL problem can be solved using an adversarial player that chooses perturbations. Rather than the worst-case, Pinto et al. (2017) optimized for the α-quantile of reward, using a conditional value-at-risk (CVAR) interpretation of adversarial robustness. We consider reward perturbations. Other work has studied perturbing transition matrices, observation spaces, and action probabilities, which are relevant to robotics and control applications (Pinto et al., 2017; Tessler et al., 2019; Hou et al., 2020; Li et al., 2019). The robust RL literature has not considered the nested RL setting at the scale of our work.

In the multi-agent setting, agents do not necessarily optimize for the same (robustness) objective. We use *one-sided robustness* in a multi-level setting: only the planner aims to be robust. This setting is significantly more difficult: previous work mostly assumed all agents act robustly (Li et al., 2019).

See Appendix A.2 for additional related work.

6 Future Work

Future work could extend ERMAS to reality gaps due to different causes, such as changes in transition dynamics, agent affordances, number of agents, etc, which might require more refined approximations to the planner's robustness objective. Furthermore, it would be interesting to apply ERMAS

to problems that feature more general solution concepts, e.g., Bayesian Nash equilibria, correlated equilibria, Walrassian equilibria, and others.

REFERENCES

Dirk Bergemann and Stephen Morris. Robust mechanism design. *Econometrica*, 73(6):1771–1813, 2005. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/3598751.

Eric Bonabeau. 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. Publisher: National Academy of Sciences Section: Colloquium Paper.

Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from Simulation to Real World through Learning Deep Inverse Dynamics Model. October 2016. URL https://arxiv.org/abs/1610.03518v1.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *International Conference on Machine Learning*, pp. 1706–1715, 2019.

Tanner Fiez, Benjamin Chasnov, and Lillian J. Ratliff. Convergence of Learning Dynamics in Stackelberg Games. *arXiv*, jun 2019. URL http://arxiv.org/abs/1906.01217.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv:1703.03400 [cs]*, July 2017. URL http://arxiv.org/abs/1703.03400. arXiv: 1703.03400.

C. Gini. *Variabilità e mutabilità*. 1912.

Irina Higgins, Arka Pal, Andrei A Rusu, Loic Matthey, Christopher P Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. *arXiv preprint arXiv:1707.08475*, 2017.

John H. Holland and John H. Miller. Artificial Adaptive Agents in Economic Theory. *American Economic Review*, 81(2):365–71, 1991. URL https://econpapers.repec.org/article/aeaaecrev/v_3a81_3ay_3a1991_3ai_3a2_3ap_3a365-71.htm. Publisher: American Economic Association.

Linfang Hou, Liang Pang, Xin Hong, Yanyan Lan, Zhiming Ma, and Dawei Yin. Robust Reinforcement Learning with Wasserstein Constraint. *arXiv:2006.00945 [cs, stat]*, June 2020. URL http://arxiv.org/abs/2006.00945. arXiv: 2006.00945.

Peter Howitt. What have central bankers learned from modern macroeconomic theory? *Journal of Macroeconomics*, 34(1):11–22, March 2012. ISSN 0164-0704. doi: 10.1016/j.jmacro.2011.08.005. URL http://www.sciencedirect.com/science/article/pii/S0164070411000619.

Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12627–12637, 2019.

Kun Ho Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Cross Domain Imitation Learning. *arXiv:1910.00105 [cs, stat]*, September 2019. URL http://arxiv.org/abs/1910.00105. arXiv: 1910.00105.

Alan P. Kirman. Whom or What Does the Representative Individual Represent? *Journal of Economic Perspectives*, 6(2):117–136, June 1992. ISSN 0895-3309. doi: 10.1257/jep.6.2.117. URL https://www.aeaweb.org/articles?id=10.1257/jep.6.2.117.

Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:4213–4220, July 2019. doi: 10.1609/aaai.v33i01.33014213.

Jun Morimoto and Kenji Doya. Robust Reinforcement Learning. In T. K. Leen, T. G. Dietterich, and V. Tresp (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.

Roger B. Myerson. *Mechanism Design*, 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.

Ofir Nachum, Michael Ahn, Hugo Ponte, Shixiang Gu, and Vikash Kumar. Multi-Agent Manipulation via Locomotion using Hierarchical Sim2Real. *arXiv:1908.05224 [cs]*, October 2019. URL http://arxiv.org/abs/1908.05224. arXiv: 1908.05224.

Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms. *arXiv:1803.02999 [cs]*, October 2018. URL http://arxiv.org/abs/1803.02999. arXiv: 1803.02999.

Santiago Paternain, Luiz F. O. Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained Reinforcement Learning Has Zero Duality Gap. *arXiv:1910.13393 [cs, math, stat]*, October 2019. URL http://arxiv.org/abs/1910.13393. arXiv: 1910.13393.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, May 2018. doi: 10.1109/ICRA.2018.8460528. URL http://arxiv.org/abs/1710.06537. arXiv: 1710.06537.

Wolfgang Pesendorfer. Behavioral economics comes of age: A review essay on advances in behavioral economics. *Journal of Economic Literature*, 44(3):712–721, 2006.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust Adversarial Reinforcement Learning. *arXiv:1703.02702 [cs]*, March 2017. URL http://arxiv.org/abs/1703.02702. arXiv: 1703.02702.

Lillian J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. On the Characterization of Local Nash Equilibria in Continuous Games. *arXiv:1411.2168 [math]*, November 2014. URL http://arxiv.org/abs/1411.2168. arXiv: 1411.2168.

Andrei A. Rusu, Mel Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-Real Robot Learning from Pixels with Progressive Nets. October 2016. URL https://arxiv.org/abs/1610.04286v2.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization. *arXiv:1502.05477 [cs]*, April 2017. URL http://arxiv.org/abs/1502.05477. arXiv: 1502.05477.

Herbert A. Simon. From substantive to procedural rationality. In T. J. Kastelein, S. K. Kuipers, W. A. Nijenhuis, and G. R. Wagenaar (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.

Herbert A. Simon and Jonathan Schaeffer. The Game of Chess. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE AND PSYCHOLOGY PROJECT, December 1990. URL https://apps.dtic.mil/sti/citations/ADA225613. Section: Technical Reports.

Young-Ho Suh, Sung-Pil Woo, Hyunhak Kim, and Dong-Hwan Park. A sim2real framework enabling decentralized agents to execute MADDPG tasks. In *Proceedings of the Workshop on Distributed Infrastructures for Deep Learning*, DIDL '19, pp. 1–6, Davis, CA, USA, December 2019. Association for Computing Machinery. ISBN 978-1-4503-7037-0. doi: 10.1145/3366622.3368146. URL https://doi.org/10.1145/3366622.3368146.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

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

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. *arXiv:1703.06907 [cs]*, March 2017. URL http://arxiv.org/abs/1703.06907. arXiv: 1703.06907.

Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, and Trevor Darrell. Adapting Deep Visuomotor Representations with Weak Pairwise Constraints. November 2015. URL https://arxiv.org/abs/1511.07111v5.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C. Parkes, and Richard Socher. 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. arXiv: 2004.13332.

A APPENDIX

A.1 ERMAS ALGORITHM PSEUDOCODE

Algorithm 1: ERMAS outer loop: `Planner-OPT`.

Result: Robust planner policy π_p.
Input: Initial planner policy π_p and agent policies $\boldsymbol{\pi}$
for ($i = 1, \ldots, n$) {
  Update parameters of agent policies $\boldsymbol{\pi}$: $\boldsymbol{\theta} \leftarrow$ `Agent-Adv-Search`($\boldsymbol{\theta}$);
  Update planner policy $\pi_p \leftarrow \pi_p + \nabla_{\pi_p} J_p(\pi_p, \boldsymbol{\pi})$;
}

Algorithm 2: ERMAS inner loop: `Agent-Adv-Search`.

Result: Parameters $\boldsymbol{\theta}$ of agent policies.
Input: Planner policy π_p, reward slack ϵ, trust region radius η, Lagrange multiplier $\boldsymbol{\lambda}$, learning rate α_λ, number of warm-up steps k, number of overall training steps n_{train}, and number of inner learning steps n_{train};
Warm up agent policies $\boldsymbol{\theta}$ with k steps of vanilla policy-gradients on $-J_p(\boldsymbol{\theta}) + \sum_{i=1}^{N} \lambda_i J_i(\boldsymbol{\theta})$;
Copy agent parameters into placeholder parameters $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$;
for ($j = 1, \ldots, n_{train}$) {
  Execute $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ to accumulate batch of experiences B_0 with mean rewards $\boldsymbol{\mu}_0$;
  Apply standard policy gradient: $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}' + \nabla_{\boldsymbol{\theta}} \left(-J_p(\boldsymbol{\theta}) + \sum_{i=1}^{N} \lambda_i J_i(\boldsymbol{\theta}) \right)$;
  for ($i = 1, \ldots, N$) {
    Initialize parameters: $\theta_i^* \leftarrow \theta_i$ *# agent i will unilaterally deviate from* $\boldsymbol{\pi}_{\boldsymbol{\theta}}$;
    for ($k = 1, \ldots, n_{inner}$) {
      Update $\theta_i^* \leftarrow \theta_i^* + \text{TRPO}\left(\theta_i^*, \theta_{-i}, \pi_p, \frac{\eta}{n_{\text{inner}}}, B_{k-1}\right)$ to increase $J_i(\theta_i^*, \theta_{-i})$;
      Execute $\pi_{\theta_i^*}, \pi_{\theta_{-i}}$ to accumulate batch of experiences B_k with mean rewards $\boldsymbol{\mu}_k$;
    }
    Approximate meta-learning correction to parameters: $\theta_i' \leftarrow \theta_i' - \lambda_i \theta_i^*$;
    Update multiplier $\lambda_i \leftarrow \lambda_i - \alpha_\lambda \left(\mu_{n_{\text{inner}},i} - \epsilon_i - \mu_{0,i}\right)$;
  }
  Copy $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}'$.
}

A.2 ADDITIONAL RELATED WORK

Robust Mechanism Design Mechanism design aims to design reward functions that have desirable equilibrium features, such as stability (Nash equilibria), truthfulness (second-price auctions), and others (Myerson, 2016). Two-level RL can be seen as designing a mechanism (reward function set by a planner) with adaptive agents. As such, in the two-level RL setting ERMAS learns a robust mechanism. Robust mechanism design (Bergemann & Morris, 2005) has studied how mechanisms can be effective even when the environment is imperfectly known. However, it is often very hard to derive analytical solutions for (robust) mechanisms. Instead, Dütting et al. (2019) used deep learning to learn (close to) optimal auctions, while learning optimal taxes by the AI Economist (Zheng et al., 2020) can be seen as adaptive mechanism design using two-level RL. In this sense, ERMAS learns a robust adaptive mechanism.

Behavioral Economics Behavioral economics has studied human reward functions and how human behavior differs from that of idealized agents optimizing simple utility functions (Pesendorfer, 2006). Simon (1976) decomposed bounded rationality into substantive and procedural rationality. Substantively rational agents have incorrectly reward function estimates (e.g., humans might be more risk-averse than a simulated agent) or who require richer games to describe their them. Procedurally rational agents are irrational agents which violate rationality assumptions or do not maximize for their reward function (Simon & Schaeffer, 1990).