A Minimax-Bayes Approach to Ad Hoc Teamwork

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learning policies for Ad Hoc Teamwork (AHT) is challenging. Most standard methods choose a specific distribution over training partners, which is assumed to mirror the distribution over partners after deployment. Moreover, they offer limited guarantees over worst-case performance. To tackle the issue, we propose using a worst-case prior distribution by adapting ideas from minimax-Bayes analysis to AHT. We thereby explicitly account for our uncertainty about the partners at test time. Extensive experiments, including evaluations on coordination tasks from the Melting Pot suite, show our method's superior robustness compared to self-play, fictitious play, and best response learning w.r.t. policy populations. This highlights the importance of selecting an appropriate training distribution over teammates to achieve robustness in AHT.

1 Introduction

Domain generalisation is often crucial in Reinforcement Learning (RL) and is typically assessed by placing an agent in novel environments (Cobbe et al., 2019). Likewise, in Multi-Agent Reinforcement Learning (MARL), generalisation to new agents can be evaluated by pairing a trained policy with unseen actors (Barrett et al., 2011; Hu et al., 2020; Leibo et al., 2021; Agapiou et al., 2023). While zero-shot domain adaptation is a valuable property (Higgins et al., 2017; Schäfer, 2022), it is equally important to ensure proper transfer to new behaviours in multi-agent settings, especially in situations where undesired interactions may arise (Gleave et al., 2019). More specifically, Ad Hoc Teamwork (AHT) occurs when multiple agents, initially unfamiliar with each other, must collaborate to achieve a common goal. In a world where autonomous agents are being progressively introduced in such tasks, cooperation with humans is becoming a major concern (Stone et al., 2010; Ji et al., 2023).

Efforts in AHT have primarily focused on learning and inferring models of teammates' behaviours (Barrett et al., 2011; Albrecht et al., 2015; Barrett et al., 2017; Chen et al., 2020; Muglich et al., 2022b), adapting to behaviour shifts (Ravula et al., 2019), and enhancing generalisation by encouraging diversity in partners during training (Jaderberg et al., 2019; Hu et al., 2020; Charakorn et al., 2020; Lupu et al., 2021; Strouse et al., 2021). However, these methods provide limited guarantees regarding worst-case AHT performance.

A multi-agent system can encompass numerous and diverse *scenarios*, each characterised by its actors. For example, autonomous cars operate alongside various human drivers and other autonomous vehicles. Similarly, in a surgical setting, a robot may need to cooperate with surgeons who have a wide range of habits and expertise levels. In each of these scenarios, we can adopt the perspective that the *focal* actors are controlled by the learner, whereas the other actors are viewed as fixed, forming the *background* of the task (Leibo et al., 2021; Agapiou et al., 2023). These scenarios can be viewed as distinct single-agent environments, as each combination of background actors induces different transition dynamics and reward functions. A common practice involves constructing representative scenarios and training a policy on a uniform distribution over them (Strouse et al., 2021; Lupu et al., 2021). However, this only ensures good performance for that specific distribution.

Submitted to 17th European Workshop on Reinforcement Learning (EWRL 2024). Do not distribute.

Recent studies in zero-shot domain transfer showed that selecting an appropriate prior over training environments is key to learning robust policies (Pinto et al., 2017; Dennis et al., 2020; Garcin et al., 2023; Jiang et al., 2021; Buening et al., 2023). Intuitively, this insight should apply to the AHT setting as well, suggesting that choosing a specific prior over scenarios/partners may improve the robustness of learned policies. Assuming that no information is available about the teammates at test time (and their distribution), we consider the *worst* possible prior over the set of partners given our policy, an idea adopted from the minimax-Bayes concept (Berger, 1985).

Contributions. We summarise our contributions as follows:

1. We adapt Minimax-Bayes Reinforcement Learning (MBRL)(Buening et al., 2023) to the AHT setting, reasoning about uncertainty with respect to partners rather than environments (Section 4).
2. We examine the advantages of using utility and regret to measure performance in the AHT setting, and propose algorithms to target either metric (Section 5).
3. We adapt a Gradient Descent-Ascent (GDA) (Lin et al., 2020) based algorithm, in conjunction with policy-gradient methods, and discuss its convergence guarantees for softmax policies (Section 6).
4. We conduct extensive experiments to evaluate our approach. We test learned policies on both seen and held-out scenarios for various cooperative problems, including partially observable games such as environments from the Melting Pot suite (Leibo et al., 2021; Agapiou et al., 2023). We compare our approach against Self-Play (SP), Fictitious Play (FP)(Brown, 1951; Heinrich et al., 2015) as well as learning a policy on a fixed uniform distribution over scenarios (Lupu et al., 2021), which is closely related to Fictitious Co-Play (FCP) (Strouse et al., 2021), as both learn the best response to population of policies (Section 7).
5. Our results confirm the theory and empirically demonstrate that our approach leads to the most robust solutions for both simple and deep RL coordination tasks, even when teammates are adaptive. This highlights the importance of choosing an appropriate distribution over training scenarios to develop policies that better transfer to new teammates.

2 Related Work

Ad Hoc Teamwork. In AHT, we are interested in developing agents capable of cooperating with other unfamiliar agents without any form of prior coordination (Rovatsos and Wolf, 2002; Stone et al., 2010; Barrett et al., 2011, 2017). Popular approaches usually involve some form of Population Play (PP), where policies forming a population are learning by interacting with each other (Lupu et al., 2021; Muglich et al., 2022a; Leibo et al., 2021; Agapiou et al., 2023). Key strategies for ensuring generalisation to new partners include promoting policy diversity within the training population (Charakorn et al., 2020) and preventing overfitting to training partners (Lanctot et al., 2017). Both Lupu et al. (2021) and Strouse et al. (2021) previously showed that learning a best response to a more diverse population leads to improved generalisation. Additionally, Jaderberg et al. (2019) showed the effectiveness of PP when diversity is encouraged through evolving pseudo-rewards. However, PP still struggles with producing policies that are robustly collaborative with new partners and sometimes exhibits overfitting (Carroll et al., 2019; Leibo et al., 2021; Agapiou et al., 2023).

To push the boundaries of AHT further, many studies use inference on teammate models to maintain a belief about ad hoc partners based on previous interactions within an episode (Barrett et al., 2011; Albrecht et al., 2015). Efforts have also been made to improve the learning and generalisation of such models to new partners (Barrett and Stone, 2015; Barrett et al., 2017; Muglich et al., 2022b).

An alternative approach proposed by Li et al. (2019) involves a robust formulation of deep deterministic policy gradients, assuming worst-case teammates. Unlike our setup, they train a joint policy that remains consistent throughout learning, and design their algorithm specifically for deep deterministic policy gradients, while our approach is compatible with any policy-gradient algorithm.

Even though the aforementioned methods attempt at improving cooperative robustness, they always assume specific distributions for the partners. Jaderberg et al. (2019) used a distribution favoring the matchmaking of policies of similar levels under the intuition that the reward signal is stronger in those cases, it does not provide any insights on its eventual effects on AHT robustness. As such, the actual impact of training partner distribution on robustness is left under-explored and represents a component that can be further exploited in conjunction with other AHT mechanisms.

Zero-shot Domain Transfer. Robustness to unknown partners can be seen as a form of zero-shot domain transfer. Each possible team composition involving the agent of interest can be considered a different environment. In the single agent setting, Jiang et al. (2021) demonstrated that adapting the training environment distribution by prioritising environments with higher prediction loss (a measure of the policy's lack of knowledge) leads to improved sample efficiency and generalisation. Building on this idea, Garcin et al. (2023) prioritised environments where the mutual information between the learning policy's internal representation and the environment identity was lower, using information theory to achieve similar results. The idea of tempering with the environment distribution was also explored by Pinto et al. (2017), who employed a maximin utility formulation to choose continuous adversarial environment perturbations throughout learning. Instead of utility, Dennis et al. (2020) stressed the advantages of using regret by proposing a training environment sampling scheme avoiding entirely unsolvable and uninformative environments. Most interestingly, Buening et al. (2023) conducted a study over worst-case priors (for both utility and regret) over training environments, and proved that worst-case distributions are a good fit for domain transfer. Finally, there exist works on domain transfer in the MARL setting (Schäfer, 2022), but this differs from our focus on transferring to new partners. This related work is consistently in favor of caring about environment distributions for robustness, providing strong motivation to bring this concern to AHT.

3 Problem Formulation

3.1 Preliminaries

An m-player Partially Observable Markov Game (POMG) is given by a tuple $\mu = \langle \mathcal{S}, \mathcal{X}, \mathcal{A}, \mathcal{O}, P, \rho, T \rangle$ defined on finite sets of states $\mathcal{S}$, observations $\mathcal{X}$ and actions $\mathcal{A}$. The observation function $\mathcal{O} : \mathcal{S} \times \{1, \dots, m\} \to \mathcal{X}$ provides a state space view for each player. In each state, each player i chooses an action $a_i \in \mathcal{A}$. Following their joint action $\mathbf{a} = (a_1, \dots, a_m) \in \mathcal{A}^m$, the state is updated according to the transition function $P : \mathcal{S} \times \mathcal{A}^m \to \Delta(\mathcal{S})$. After a transition, each player receives a reward defined by $\rho : \mathcal{S} \times \mathcal{A}^m \times \{1, \dots, m\} \to \mathbb{R}$. The game ends after T transitions. Permuting player indices does not have any effect on μ.

A policy $\pi : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \times \mathcal{A} \times \cdots \times \mathcal{X} \to \Delta(\mathcal{S})$ is a probability distribution over a single agent's actions, conditioned on that agent's history of observations and actions. We denote Π the set of all policies and $\Pi^{\mathrm{D}} \subset \Pi$ the set of deterministic policies.

3.2 Scenarios

Let a *scenario* σ_b^c be defined by its number of *focal* players c, and its *background* players $\boldsymbol{\pi}^b = (\pi_1^b, \dots, \pi_{m-c}^b) \in \Pi^{m-c}$. We say we deploy a policy π^f in scenario σ_b^c if the c focal players are set to copies of π^f. Hence, in addition to the $m - c$ many background policies $\boldsymbol{\pi}^b$, there are c many focal policies $\boldsymbol{\pi}^f = (\pi^f, \dots, \pi^f)$. We sometimes use σ as a shorthand notation for σ_b^c for simplicity. We also denote $\mathbf{a}^f \in \mathcal{A}^c$ and $\mathbf{a}^b \in \mathcal{A}^{m-c}$ the joint actions of the focal and background players, respectively. A background population $\mathcal{B} \subset \Pi$ is a finite set of policies, to which we assign a set of scenarios:

$$\Sigma(\mathcal{B}) \triangleq \{\sigma_b^c \mid 1 < c \leq m, \boldsymbol{\pi}^b \in \mathcal{B}^{m-c}\}.$$

A scenario σ_b^c on μ can be viewed as its own c-player POMG, through the marginalisation of the policies of its background players.[1] We denote $\mu(\sigma) = \langle \mathcal{S}, \mathcal{X}, \mathcal{A}, \mathcal{O}_\sigma, P_\sigma, \rho_\sigma, \gamma, T \rangle$ the POMG induced by scenario σ, where $\mathcal{O}_\sigma : \mathcal{S} \times \{1, \dots, c\} \to \mathcal{X}$ is the corresponding observation function, $P_\sigma : \mathcal{S} \times \mathcal{A}^c \to \Delta(\mathcal{S})$ the transition function given by

$$P_\sigma(s' \mid s, \mathbf{a}^f) = \begin{cases} P(s' \mid s, \mathbf{a}^f), & c = m \\ \sum_{\mathbf{a}^b} \left(P(s' \mid s, \mathbf{a}^f, \mathbf{a}^b) \prod_i \boldsymbol{\pi}_i^b(\mathbf{a}_i^b \mid h_i) \right), & c < m \end{cases}$$

and $\rho_\sigma : \mathcal{S} \times \mathcal{A}^c \times \{1, \dots, c\} \to \mathbb{R}$ the induced reward function with:

$$\rho_\sigma(s, \mathbf{a}^f, i) = \begin{cases} \rho(s, \mathbf{a}^f, i), & c = m \\ \sum_{\mathbf{a}^b} \left(\rho(s, \mathbf{a}^f, \mathbf{a}^b, i) \prod_i \boldsymbol{\pi}_i^b(\mathbf{a}_i^b | h_i) \right), & c < m \end{cases}$$

[1] Each scenario can be seen as a decentralized partially observable Markov decision process (Oliehoek, 2012) constrained by the fact that the c players are copies.

where h_i is the history of observations and actions of the i-th policy and $\mathbf{a}_i^b$ its action in $\mathbf{a}^b$. We denote the scenario that only involves copies of the focal policy, i.e. the universalisation scenario (Leibo et al., 2021), by $\sigma^{\text{SP}} = \sigma_\emptyset^m$.

3.3 Evaluation

The expected utility of a policy in scenario σ is the mean return of the focal policies given by the expected *focal-per-capita return* (Leibo et al., 2021; Agapiou et al., 2023):

$$U(\pi, \sigma) \triangleq \sum_{t=1}^{T} \frac{1}{c} \sum_{i=1}^{c} \mathbb{E}^{\pi}_{\mu(\sigma)}[\rho_\sigma(s_t, \mathbf{a}_t^f, i)]. \tag{1}$$

$U^*(\sigma) \triangleq \max_{\pi \in \Pi} U(\pi, \sigma)$ denotes the maximal utility achievable in scenario σ. This definition for utility represents the need for autonomous agents to always maximise the mean joint rewards of its copies, regardless of the scenario. We can further define the notion of regret incurred by deploying some policy π on scenario σ, as the gap between the maximal utility and the utility of π on σ:

$$R(\pi, \sigma) \triangleq U^*(\sigma) - U(\pi, \sigma). \tag{2}$$

To assess a learning method in terms of AHT, we use the evaluation protocol of Leibo et al. (2021). This has two phases:

1. **Training phase**: A test background population $\mathcal{B}^{\text{test}}$ is kept hidden. The policy learner has access to the game μ with no restriction, beside accessing $\mathcal{B}^{\text{test}}$. For example, the learner is free to use a modified instance μ' of μ, where $\mathcal{O}$ could be changed to include observations of other players, or again where ρ could be tweaked to return the joint rewards rather than individual rewards.
2. **Testing phase**: The obtained policy is frozen and cannot be trained any further. We compute the performance of the policy on μ by taking its average expected utility across a series of held-out test scenarios $\Sigma^{\text{test}} \subset \Sigma(\mathcal{B}^{\text{test}})$. In addition to performance, we consider two metrics related to robustness, worst-case utility and worst-case regret:

$$p(\pi, \Sigma) = \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} U(\pi, \sigma), \qquad U^-(\pi, \Sigma) = \min_{\sigma \in \Sigma} U(\pi, \sigma), \qquad R^+(\pi, \Sigma) = \max_{\sigma \in \Sigma} R(\pi, \sigma). \tag{3}$$

   Maximising U^- is typically preferable when falling below a certain utility threshold must be avoided at all costs; for instance minimising casualties in a surgical context. Conversely, minimising R^+ avoids decisions that lead to significantly worse outcomes than the best-case.

The end objective is to design a learning process outputting a policy that reliably maximises its expected utility (focal-per-capita return) on possibly unseen scenarios.

3.4 Assumptions

To ensure our setting aligns with the AHT literature, we must adhere to three assumptions (Mirsky et al., 2022): a) the absence of prior coordination. The learner must be capable of cooperating with the team on-the-fly, without relying on previously established collaboration strategies, even between copies of the learner's agent. b) There is no control over teammates, the learner can control its own copies but not other agents in the configuration. c) All agents are assumed to share a common objective. Nonetheless, their reward function may be different, reflecting varying preferences. In this work, we choose to address this last point by assuming a class of possible reward functions for the background players. In an attempt to model realistic situations, we formalise this diversity by considering various levels of prosociality λ (Peysakhovich and Lerer, 2017) and risk-aversion δ for each policy. To illustrate with an example, in a setting where company coworkers have to realise a project, there might be workers that have a high preference over their own contribution (with a better chance to get promoted later), while there may be others that are inclined to delegate their work for things they are unsure about to the team:

$$\rho_{\text{social+risk}}(s, \mathbf{a}, i) = \rho^+_{\text{social}}(s, \mathbf{a}, i) - \delta_i \rho^-_{\text{social}}(s, \mathbf{a}, i),$$

with ρ_{social} defined as

$$\rho_{\text{social}}(s, \mathbf{a}, i) = \lambda_i \rho(s, \mathbf{a}, i) + (1 - \lambda_i) \sum_{j=1}^{m} \rho(s, \mathbf{a}, j),$$

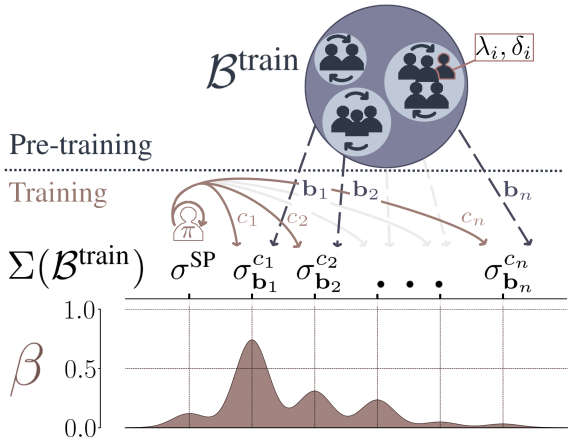


Figure 1: Comprehensive illustration of the framework used in this paper. Prior to training the focal policy π, background policies with different preferences (λ_i, δ_i) learn by interacting within sub-populations of varying sizes. These sub-populations are then combined to form a background population, $\mathcal{B}^{\text{train}}$, used as a common 'train dataset' for all algorithms.

Our primary focus is on the training phase, where the main policy π is learned alongside the distribution β over scenarios. These scenarios mix copies of π with policies from $\mathcal{B}^{\text{train}}$, where the self-play scenario σ^{SP} has the policy interacting only with copies of itself.

where f^+ and f^- are the positive and negative parts of f, and (λ_i, δ_i) are the levels of prosociality and risk-aversion for agent i. Combining values of prosociality and risk-aversion allows for the consideration of behaviours with a wide range of preferences.

4 Achieving Robust AHT

To learn a policy able to cooperate with new partners, a straightforward idea is to reconstruct scenarios that would likely be encountered in nature. A roadblock to this approach however is that it requires two main ingredients: a) a diverse pool of partners, and b) a prior distribution over them. The prior, often neglected, is important as it captures our uncertainty about the true partners observed in nature.

In Section 4.1, we reflect on motivating previous work on diverse behaviour generation, before describing our own adopted approach. Section 4.2 then introduces the Minimax-Bayes idea to AHT, by stating the existing connections of the setting with MBRL's.

4.1 Constructing Training Scenarios

Prior to learning any robust policy, we need to construct diverse scenarios. A background population that encompasses a wide range of behaviours is needed. Previous work on AHT tackled the issue in various manners, such as using genetic algorithms (Muglich et al., 2022b), rule-based policies generated with MAP Elites (Canaan et al., 2023), SP policies (Strouse et al., 2021), explicit behavior diversification through regularisation (Lupu et al., 2021), or through evolved pseudo-rewards (Jaderberg et al., 2019). Based on real-life examples and aiming to thoroughly assess the effects of partner priors, we adopt the following approach:

- Each background policy has unique preferences (λ_i, δ_i).
- Policies are organized into sub-populations $\mathcal{B} = \bigcup_k \mathcal{B}_k$ of varying sizes, simulating different communities.
- Each sub-populations are separately trained using PP. Given the diverse preferences and varying sizes of these sub-populations, distinct habits, common practices, and established conventions will emerge within each group, effectively mimicking various cultures.

This choice for constructing scenarios is rather arbitrary and is not the main focus of our work. Nevertheless, a rigorous generation procedure is important to bring forward the effects of various scenario priors on AHT robustness.

4.2 Minimax-Bayes AHT

In the standard single-agent Bayesian RL setting, the learner selects a subjective belief β over candidate Markov Decision Processes (MDPs) $\mathcal{M}$ for the unknown, true environment $\mu^* \in \mathcal{M}$. The learner's objective is to maximise its expected expected utility with respect to the chosen prior $U(\pi, \beta) = \int_{\mathcal{M}} U(\pi, \mu) \, \mathrm{d}\beta(\mu)$, i.e. finding the Bayes-optimal policy. In MBRL, Buening et al. (2023) proposed considering the worst possible prior for the agent, without knowledge of the policy that will be chosen. This approach can be interpreted as nature playing the minimising player against the policy learner in a simultaneous-move zero-sum normal-form game. Learning against a worst-case prior intuitively makes the learner more robust, as it prepares for the worst outcomes.

To transfer this idea to our setting, we remark that any finite population $\mathcal{B}$ provides a finite set of POMGs $\mathcal{M}_{\mathcal{B}} = \{\mu(\sigma) | \sigma \in \Sigma(\mathcal{B})\}$. The difference here is the use of POMGs rather than MDPs. We extend the notion of expected utility with respect to a prior over scenarios, i.e. when $\beta \in \Delta(\Sigma(\mathcal{B}))$:

$$U(\pi, \beta) \triangleq \mathbb{E}_{\sigma \sim \beta}[U(\pi, \sigma)] = \sum_{\sigma} U(\pi, \sigma) \beta(\sigma).$$

This allows us to formulate the following maximin game:

$$\max_{\pi \in \Pi} \min_{\beta \in \Delta(\Sigma(\mathcal{B}))} U(\pi, \beta). \tag{4}$$

Similarly to Buening et al. (2023), we are interested in knowing whether such a game has a solution (i.e., a value), assuming that nature and the agent play simultaneously without knowledge of each other's move. This is relevant in our setting because the policy learner does not know the true distribution of partners available in nature, while the effective nature's distribution of scenarios should not depend on the agent's policy. Fortunately, (4) has a value when $\mathcal{B}$ is finite.

Corollary 1. *For an m-player POMG μ in a finite state-action space, with a known reward function and a finite horizon, and a finite background population $\mathcal{B}$, the maximin game* (4) *has a value:*

$$\max_{\pi \in \Pi} \min_{\beta \in \Delta(\Sigma(\mathcal{B}))} U(\pi, \beta) = \min_{\beta \in \Delta(\Sigma(\mathcal{B}))} \max_{\pi \in \Pi} U(\pi, \beta). \tag{5}$$

Proof. First, observe that for any stochastic policy $\pi \in \Pi$, there exists a distribution over deterministic policies $\phi \in \Delta(\Pi^{\mathrm{D}})$ such that $\pi(a_t | h_t) = \sum_{d \in \Pi^{\mathrm{D}}} d(a_t | h_t) \phi(d)$. Consequently, we can rewrite the utility as $U(\pi, \beta) = \sum_{d \in \Pi^{\mathrm{D}}} \sum_{\sigma \in \Sigma(\mathcal{B})} U(d, \sigma) \phi(d) \beta(\sigma)$. This demonstrates that U is bilinear in ϕ and β, which allows us to apply the minimax theorem, thus proving the result. □

Importantly, prior work that chooses an arbitrarily fixed prior is limited in terms of robustness guarantees: it only ensures maximal utility for their specific prior. In contrast, a policy π^*_U solving the maximin utility problem (4) has its expected utility lower-bounded on $\Sigma(\mathcal{B})$:

$$\forall \beta \in \Delta(\Sigma(\mathcal{B})), \quad U(\pi^*_U, \beta) \geq U(\pi^*_U, \beta^*_U), \tag{6}$$

where β^*_U is the worst-case prior for π^*_U. Simply put, π^*_U performs the worst when the prior is its worst-case β^*_U, but can only improve when the prior deviates from β^*_U. Additionally, it is also optimal on the worst-case prior:

$$\forall \pi \in \Pi, \quad U(\pi^*_U, \beta^*_U) \geq U(\pi, \beta^*_U). \tag{7}$$

Note that is is entirely different from the best response to the fixed worst-case prior $\arg\max_{\pi} U(\pi, \beta^*_U)$, which once again, only has a guaranteed optimal utility on β^*_U.

5 Utility or Regret?

Optimising for the worst-case utility (4) might be problematic. Nature could resort to only picking scenarios where the the focal players achieve the worst possible score. Then, the prior trivially minimises utility for any chosen policy. Buening et al. (2023) addresses this issue by instead considering the regret of a policy.. The difference is that 'impossible' scenarios will always yield zero regret for any policy, thus becoming irrelevant for a regret-maximising nature. Letting $L(\pi, \beta) \triangleq \int_{\mathcal{M}} R(\pi, \mu) \, \mathrm{d}\beta(\mu)$ be the Bayesian regret with respect to a prior β, we now formulate the following minimax regret game:

$$\min_{\pi \in \Pi} \max_{\beta \in \Delta(\Sigma(\mathcal{B}))} L(\pi, \beta). \tag{8}$$

One can also prove that this above game has a value. Additionally, a solution (π_R^*, β_R^*) solving (8) has similar properties to (6) and (7) with respect to regret: π_R^* has its Bayesian regret upper bounded by $L(\pi_R^*, \beta_R^*)$ on $\Sigma(\mathcal{B})$ and is optimal on β_R^*.

Should utility or regret be used as an objective? Exploiting Regret ensures that scenarios from which you can learn the most from are sampled more often. It also ensures that degenerate scenarios get discarded as their regret is always zero. However, it demands the calculation of best responses for each scenario, which becomes taxing as the number of scenarios or problem complexity grows. To reduce the computational burden, we can approximate those best responses, or subsample the set of scenarios. An alternative way is to make use of the utility notion under some additional conditions:

Definition 1 (Non-degenerative population)**.** A background population of policies $\mathcal{B} \subset \Pi$ is non-degenerative $\iff \forall \sigma \in \Sigma(\mathcal{B}), \exists \pi_1, \pi_2 \in \Pi, \pi_1 \neq \pi_2$ and $U(\pi_1, \sigma) \neq U(\pi_2, \sigma)$.

Lemma 1. *If a population $\mathcal{B}$ is non-degenerative, then $\forall \sigma \in \Sigma(\mathcal{B}), \exists \pi \in \Pi, R(\pi, \sigma) > 0$.*

Proof. $\mathcal{B}$ is non-degenerative, for any scenario $\sigma \in \Sigma(\mathcal{B})$ there must exist two policies π_1 and π_2 such that $U(\pi_1, \sigma) > U(\pi_2, \sigma)$. We have by definition $U^*(\sigma) \geq U(\pi_1, \sigma)$, hence $R(\pi_2, \sigma) > 0$. □

Making the assumption that a background population is non-degenerative is in general realistic for cooperative tasks. This translates into only considering reasonable behaviors for the background population, or tasks where teammates cannot completely cancel out the actions of the focal players. Under the assumption of a non-degenerative population, no distribution can deadlock the policy learner into a stale scenario. For the remainder of the paper, background populations are assumed to be non-degenerative.

6 Computing Solutions

We desire to calculate the solution pairs for both the maximin utility (4) and minimax regret (8) games. Buening et al. (2023) theoretically proved that GDA has convergence guarantees when the game is played between a policy learned with softmax parameterization and nature learning its distribution over a finite set of MDPs. These results apply if all scenarios induce single-agent POMGs, as partial observability does not interfere with proving the required properties. However, when the focal policy is deployed in a scenario with $c > 1$ copies, the game is no longer single-agent.

To approximate the reduction of these multi-agent POMGs to single-agent POMGs during training, we propose using delayed versions π_{t-d} of the focal policy π_t for the $c - 1$ remaining copies. This common practice smooths the behavior of the copies and favors proper convergence by treating the copies as fixed policies. An implementation of GDA for our setting is provided in Appendix A.

7 Experiments

The aim of our experiments is to highlight the importance of partner distribution in the learning process. To achieve this, we evaluate our proposed strategies, Maximin Utility (MU) and Minimax Regret (MR), on two distinct problems. First, we consider the fully known and observable repeated Prisoner's Dilemma to validate the theoretical results. Following this, we test our approaches on a deep-learning task, the Collaborative Cooking (Overcooked) game (Carroll et al., 2019; Strouse et al., 2021; Leibo et al., 2021; Agapiou et al., 2023). Throughout our experiments, we benchmark MU and MR against other distribution management strategies: SP which fixes the prior as the Dirac distribution $\beta^{\text{SP}}(\sigma^{\text{SP}}) = 1$, FP which is similar to SP but has the versions of the copies sampled uniformly from the full history of policies $\pi_0, \ldots, \pi_t$, and Population Best-Response (PBR) which learns the best response to the training background population by maintaining a uniform prior $\beta^{\text{PBR}} = \mathcal{U}(\Sigma(\mathcal{B}^{\text{train}}))$. Approaches are consistently evaluated on their training background population, as well as on a separate test set, in order to evaluate their AHT capabilities.

7.1 Repeated Prisoner's Dilemma

In these experiments, all computations can be exact. This includes the gradient calculation for the prior, as well as for the agent's policies. We focus on the repeated Prisoner's Dilemma, where two players play the matrix game repeatedly for $T = 3$ rounds.

Table 1: Scores on the repeated Prisoner's Dilemma. A higher value is desired for performance (p, average utility) and worst-case utility (U^-), while a lower worst-case regret (R^+) is better. $p(\beta)$ corresponds to the utility w.r.t. a specific distribution β, rather than the average as in (3).

	$\Sigma(\mathcal{B}^{\text{train}})$					$\Sigma(\mathcal{B}^{\text{test}})$		
	p	$p(\beta^*_U)$	$p(\beta^*_R)$	U^-	R^+	p	U^-	R^+
MU	6.82	**3.00**	8.10	**3.00**	8.92	**8.65**	**3.08**	8.92
MR	9.14	0.92	11.25	0.92	**2.11**	7.54	0.99	8.45
PBR	**10.0**	1.96	10.66	1.96	3.00	7.74	1.99	10.46
FP	9.69	0.14	11.89	0.14	2.95	7.10	0.17	10.53
SP	9.69	0.46	**11.99**	0.46	3.00	7.21	0.47	10.70
TfT	**10.0**	2.00	11.73	2.00	3.00	7.60	2.01	10.65
CuD	**10.0**	2.00	11.73	2.00	3.00	7.85	2.01	9.92
Random	8.10	1.50	10.10	1.5	4.5	8.00	2.52	**4.5**

Experimental Setup. In the repeated Prisoner's Dilemma, players receive and observe rewards based on their chosen actions: players receive a reward of 1 if both defect, 4 if both cooperate, 5 and 0 if the first defects while the second cooperates. The game has one state, and the outcomes observed are enough to determine the joint actions, making it fully observable.

We use softmax, fully adaptive policies, where actions depend on the entire history of observations and actions. During training, the learner has access to a background population containing a pure cooperative policy, a pure defective policy, and two popular strategies for the game: Tit-for-Tat (TfT), which mimics the partner's previous action and starts with a cooperate action, and Cooperate-until-Defected (CuD), which defects if any defection was observed in the past, otherwise cooperating. A separate test population $\mathcal{B}^{\text{test}}$ is generated beforehand by randomly sampling 32 stochastic policies.

Results. Table 1 presents scores on the training and test sets. The results on the training set confirm most expectations: PBR is the best under the uniform prior (p), MU has the highest worst-case utility (U^-), and MR has the lowest worst-case regret (R^+). However, MR is not optimal on the worst-case prior β^*_R, likely due to the approximate self-play. On the test set, the best performance is achieved by MU, rather than PBR. MU also has the highest worst-case utility. Lastly, the random strategy excels in terms of worst-case regret on the test set, likely because it avoids fully committing to defecting or cooperating. Besides the random strategy, MR has the most respectable worst-case regret, closely followed by MU. These results indicate that learning the best response to populations does not ensure the best robustness to new partners. We also remark that SP and FP agents seem to overfit to their own established conventions, resulting in poor transferability to training and test policies.

7.2 Robust Cooperation in Deep RL Tasks

For this section, we tackle the Collaborative Cooking game (Agapiou et al., 2023), where two players act as chefs in a gridworld kitchen, working together to deliver as many tomato soup dishes as possible within a set time. To do so, they have to collect tomatoes, cook them, prepare dishes, and deliver the soup. Successful deliveries reward both players equally. Players must navigate the kitchen, interact with objects in the right order, and coordinate with each other. Each player has an egocentric, partial RGB view of the environment. All of our policies in this section are using deep recurrent (LSTM) neural networks.

Experimental Setup. Two separate background populations, $\mathcal{B}^{\text{train}}$ and $\mathcal{B}^{\text{test}}$, are generated according to Section 4.1. Both populations are trained with an identical setup, differing only in their seed. Each is partitioned into four sub-populations of sizes 2, 3, and 5, totaling 10 policies. Prosociality and risk-aversion are sampled uniformly in $[-0.2, 1.2]$ and $[0.1, 2]$ respectively. The same populations are used throughout three random seeds.

For a fair comparison and to focus on scenario distribution learning, we assume that $\mathcal{B}^{\text{train}}$ is readily available to all approaches, which can be exploited for a maximum of 4×10^7 environment steps to learn a policy with PPO (Schulman et al., 2017). We evaluate the approaches on two different kitchen layouts: Circuit and Cramped (Agapiou et al., 2023).

Table 2: Scores on the Collaborative Cooking environment training set. The standard error is taken over three random seeds. The scores are aggregated over kitchen layouts.

	$\Sigma(\mathcal{B}^{\text{train}})$				
	p	$p(\beta_U^*)$	$p(\beta_R^*)$	U^-	R^+
MU	$\mathbf{266.9 \pm 4.3}$	$\mathbf{23.1 \pm 0.7}$	$\mathbf{24.5 \pm 0.8}$	$\mathbf{225.3 \pm 11.5}$	266.0 ± 7.9
MR	232.0 ± 18.6	19.9 ± 1.6	20.2 ± 1.4	144.3 ± 14.4	$\mathbf{230.7 \pm 28.2}$
PBR	209.7 ± 23.9	20.0 ± 1.7	18.4 ± 3.1	96.8 ± 13.4	357.6 ± 16.1
FP	129.9 ± 13.9	7.9 ± 0.7	12.0 ± 1.1	0.2 ± 0.1	483.5 ± 16.1
SP	124.8 ± 26.4	9.4 ± 2.8	11.8 ± 3.2	15.7 ± 10.5	460.8 ± 21.7
Random	42.8 ± 0.0	6.7 ± 0.0	4.2 ± 0.0	0.0 ± 0.0	505.4 ± 0.0

Table 3: Scores on the Collaborative Cooking environment test sets.

	$\Sigma(\mathcal{B}^{\text{test}})$			$\Sigma^{\text{Melting Pot}}$		
	p	U^-	R^+	p	U^-	R^+
MU	$\mathbf{195.7 \pm 6.2}$	$\mathbf{66.0 \pm 6.8}$	266.4 ± 10.1	$\mathbf{273.8 \pm 4.9}$	$\mathbf{224.9 \pm 7.1}$	$\mathbf{118.0 \pm 7.1}$
MR	172.2 ± 15.4	65.1 ± 16.0	$\mathbf{248.2 \pm 28.4}$	206.8 ± 12.6	148.7 ± 9.1	187.1 ± 13.0
PBR	151.4 ± 19.5	33.6 ± 5.9	327.1 ± 14.0	171.8 ± 21.1	106.3 ± 9.3	228.1 ± 5.8
FP	152.2 ± 16.8	16.7 ± 11.7	369.7 ± 19.7	121.5 ± 15.7	40.7 ± 16.3	294.8 ± 10.7
SP	117.4 ± 12.5	6.7 ± 3.5	367.5 ± 11.6	101.2 ± 17.8	29.0 ± 17.4	293.4 ± 14.5
PP-ACB	n/a	n/a	n/a	82.4 ± 0.0	0.0 ± 0.0	307.3 ± 0.0
PP-OPRE	n/a	n/a	n/a	102.3 ± 0.0	14.6 ± 0.0	292.7 ± 0.0
PP-VMPO	n/a	n/a	n/a	78.6 ± 0.0	36.1 ± 0.0	306.7 ± 0.0
Random	32.2 ± 0.0	0.0 ± 0.0	445.0 ± 0.0	60.6 ± 0.0	0.0 ± 0.0	307.3 ± 0.0

Additionally, we assess the learned policies on the Melting Pot benchmark scenarios, comparing them against the scores of the baselines reported in the original paper (Agapiou et al., 2023): an Actor-Critic Baseline (ACB), V-MPO (Song et al., 2019), and OPRE (Vezhnevets et al., 2020). Note that these three baselines were trained through PP without our background policies, for 10^9 environment steps.

Results. The results in Table 2 and Table 3 clearly show that MU marginally outperforms all other evaluated methods. Looking at the robustness metrics, MU has the best worst-case utilities (U^-) and the best worst-case regret (R^+) on the Melting Pot scenarios. MR also performs better than any other benchmarked method overall, securing the lowest worst-case regrets on both the training and test sets. In terms of performance (p), MU and MR are consistently the best and second-best, respectively, which is particularly notable on the training set where PBR was expected to perform the best.

One hypothesis for MR performing worse than MU globally is that the estimations of the maximal utilities for training scenarios are too approximate. Another hypothesis for why MU and MR marginally outperform PBR and other approaches is that the distributions learned during training have a similar effect to curriculum learning, introducing indirect exploration in behaviours compared to fixed distributions.

8 Conclusion

We investigated how to obtain robust adaptive policies in an AHT setting. Leveraging work on Minimax-Bayes RL, we proposed a method to find worst-case distributions over background populations, which led to consistently robust policies compared to simply training policies on uniform distributions. In addition, we have found the unexpected results that these distributional choices significantly accelerate learning. For future work, we believe that adapting our methods for a partner distribution-based curriculum-learning could be highly promising. This approach has the potential to strongly enhance sample efficiency, asymptotic performance, and robustness.

References

John P. Agapiou, Alexander Sasha Vezhnevets, Edgar A. Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, DJ Strouse, Michael B. Johanson, Sukhdeep Singh, Julia Haas, Igor Mordatch, Dean Mobbs, and Joel Z. Leibo. 2023. Melting Pot 2.0. arXiv:2211.13746 [cs.MA]

Stefano Albrecht, Jacob Crandall, and Subramanian Ramamoorthy. 2015. An Empirical Study on the Practical Impact of Prior Beliefs over Policy Types. *Proceedings of the AAAI Conference on Artificial Intelligence* 29, 1 (Feb. 2015).

Samuel Barrett, Avi Rosenfeld, Sarit Kraus, and Peter Stone. 2017. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence* 242 (2017), 132–171.

Samuel Barrett and Peter Stone. 2015. Cooperating with Unknown Teammates in Complex Domains: A Robot Soccer Case Study of Ad Hoc Teamwork. *Proceedings of the AAAI Conference on Artificial Intelligence* 29, 1 (Feb. 2015).

Samuel Barrett, Peter Stone, and Sarit Kraus. 2011. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. 567–574.

James O Berger. 1985. Statistical decision theory and Bayesian analysis. *Springer Series in Statistics* (1985).

George W. Brown. 1951. Iterative Solution of Games by Fictitious Play. In *Activity Analysis of Production and Allocation*, T. C. Koopmans (Ed.). Wiley, New York.

Thomas Kleine Buening, Christos Dimitrakakis, Hannes Eriksson, Divya Grover, and Emilio Jorge. 2023. Minimax-Bayes Reinforcement Learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 206)*, Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (Eds.). PMLR, 7511–7527.

Rodrigo Canaan, Xianbo Gao, Julian Togelius, Andy Nealen, and Stefan Menzel. 2023. Generating and Adapting to Diverse Ad Hoc Partners in Hanabi. *IEEE Transactions on Games* 15, 2 (2023), 228–241.

Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems* 32 (2019).

Rujikorn Charakorn, Poramate Manoonpong, and Nat Dilokthanakul. 2020. Investigating partner diversification methods in cooperative multi-agent deep reinforcement learning. In *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part V 27*. Springer, 395–402.

Shuo Chen, Ewa Andrejczuk, Zhiguang Cao, and Jie Zhang. 2020. Aateam: Achieving the ad hoc teamwork by employing the attention mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 7095–7102.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. 2019. Quantifying Generalization in Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 1282–1289.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. 2020. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems* 33 (2020), 13049–13061.

Samuel Garcin, James Doran, Shangmin Guo, Christopher G Lucas, and Stefano V Albrecht. 2023. How the level sampling process impacts zero-shot generalisation in deep reinforcement learning. *arXiv preprint arXiv:2310.03494* (2023).

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. 2019. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615* (2019).

Johannes Heinrich, Marc Lanctot, and David Silver. 2015. Fictitious Self-Play in Extensive-Form Games. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 805–813.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. 2017. DARLA: Improving Zero-Shot Transfer in Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 1480–1490.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. 2020. "Other-Play" for Zero-Shot Coordination. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 4399–4410.

Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* 364, 6443 (2019), 859–865.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852* (2023).

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. 2021. Prioritized Level Replay. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 4940–4950.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. 2017. A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

Joel Z Leibo, Edgar A Dueñez-Guzman, Alexander Vezhnevets, John P Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charlie Beattie, Igor Mordatch, and Thore Graepel. 2021. Scalable Evaluation of Multi-Agent Reinforcement Learning with Melting Pot. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 6187–6199.

Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. 2019. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 4213–4220.

Tianyi Lin, Chi Jin, and Michael Jordan. 2020. On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 6083–6093.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. 2021. Trajectory Diversity for Zero-Shot Coordination. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 7204–7213.

Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V. Albrecht. 2022. A Survey of Ad Hoc Teamwork Research. In *Multi-Agent Systems*, Dorothea Baumeister and Jörg Rothe (Eds.). Springer International Publishing, Cham, 275–293.

Darius Muglich, Christian Schroeder de Witt, Elise van der Pol, Shimon Whiteson, and Jakob Foerster. 2022a. Equivariant Networks for Zero-Shot Coordination. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 6410–6423.

Darius Muglich, Luisa M Zintgraf, Christian A Schroeder De Witt, Shimon Whiteson, and Jakob Foerster. 2022b. Generalized Beliefs for Cooperative AI. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 16062–16082.

Frans A Oliehoek. 2012. Decentralized pomdps. In *Reinforcement learning: state-of-the-art*. Springer, 471–503.

Alexander Peysakhovich and Adam Lerer. 2017. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv preprint arXiv:1709.02865* (2017).

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust Adversarial Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 2817–2826.

Manish Ravula, Shani Alkoby, and Peter Stone. 2019. Ad Hoc Teamwork With Behavior Switching Agents. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 550–556.

Michael Rovatsos and Marco Wolf. 2002. Towards social complexity reduction in multiagent learning: the adhoc approach. In *Proceedings of the 2002 AAAI Spring Symposium on Collaborative Learning Agents*. 90–97.

Lukas Schäfer. 2022. Task generalisation in multi-agent reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 1863–1865.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

H Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, et al. 2019. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control. *arXiv preprint arXiv:1909.12238* (2019).

Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 24. 1504–1509.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. 2021. Collaborating with humans without human data. *Advances in Neural Information Processing Systems* 34 (2021), 14502–14515.

Alexander Vezhnevets, Yuhuai Wu, Maria Eckstein, Rémi Leblond, and Joel Z Leibo. 2020. OPtions as REsponses: Grounding behavioural hierarchies in multi-agent reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 9733–9742.

Algorithm 1 Background-Focal GDA

Input set of background policies $\mathcal{B}$, and learning rates (η_π, η_β).
Initialize randomly the main policy parameters θ_0
Initialize the belief as the uniform distribution over possible scenarios $\beta_0 = \mathcal{U}(\Sigma(\mathcal{B}))$
for $t = 0, \dots, N-1$ **do**
  Compute $U(\pi_{\theta_t}, \sigma)$ for all $\sigma \in \Sigma(\mathcal{B})$
  Compute $U(\pi_{\theta_t}, \beta_t) = \sum_\sigma U(\pi_{\theta_t}, \sigma)\beta_t(\sigma)$
  Compute $R(\pi_{\theta_t}, \sigma) = U^*(\sigma) - U(\pi_{\theta_t}, \sigma)$ for all $\sigma \in \Sigma(\mathcal{B})$
  Obtain $L(\pi_{\theta_t}, \beta_t) = \sum_\sigma R(\pi_{\theta_t}, \sigma)\beta_t(\sigma)$
  Update belief $\beta_{t+1} = \mathcal{P}(\beta_t + \eta_\beta \nabla_\beta L(\pi_{\theta_t}, \beta_t))$ (projection onto the simplex)
  Update policy parameters $\theta_{t+1} = \theta_t + \eta_\theta \nabla_\theta U(\pi_{\theta_t}, \beta_t)$
end for
return θ^*, β^* uniformly at random from $\{(\theta_1, \beta_1), \dots, (\theta_N, \beta_N)\}$

Algorithm 2 Background-Focal SGDA

Input set of background policies $\mathcal{B}$, batch size B, learning rates (η_π, η_β)
Initialize randomly the main policy parameters θ_0
Initialize the belief as the uniform distribution over possible scenarios $\beta_0 = \mathcal{U}(\Sigma(\mathcal{B}))$
for $t = 0, \dots, N-1$ **do**
  Sample B scenarios $\sigma_1, \dots, \sigma_B \sim \beta_t$
  Estimate $\hat{U}(\pi_{\theta_t}, \sigma_i)$ by deploying π_{θ_t} on σ_i, for $i = 1, \dots, B$
  Compute $\hat{U}(\pi_{\theta_t}, \beta_t) = \frac{1}{B}\sum_{i=1}^B \hat{U}(\pi_{\theta_t}, \sigma_i)$
  Compute $\hat{R}(\pi_{\theta_t}, \sigma_i) = U^*(\sigma_i) - \hat{U}(\pi_{\theta_t}, \sigma_i)$ for each $i = 1, \dots, B$
  Compute $\hat{L}(\pi_{\theta_t}, \beta_t) = \frac{1}{B}\sum_{i=1}^B \hat{R}(\pi_{\theta_t}, \sigma_i)$
  Update belief $\beta_{t+1} = \mathcal{P}(\beta_t + \eta_\beta \nabla_\beta \hat{L}(\pi_{\theta_t}, \beta_t))$ (projection onto the simplex)
  Update policy parameters $\theta_{t+1} = \theta_t + \eta_\theta \nabla_\theta \hat{U}(\pi_{\theta_t}, \beta_t)$
end for
return θ^*, β^* uniformly at random from $\{(\theta_1, \beta_1), \dots, (\theta_N, \beta_N)\}$

Appendix

A Algorithms

We provide implementations for solving the minimax regret problem with (Algorithm 1) or without (Algorithm 2) full knowledge of the game. Updating the belief with the rule

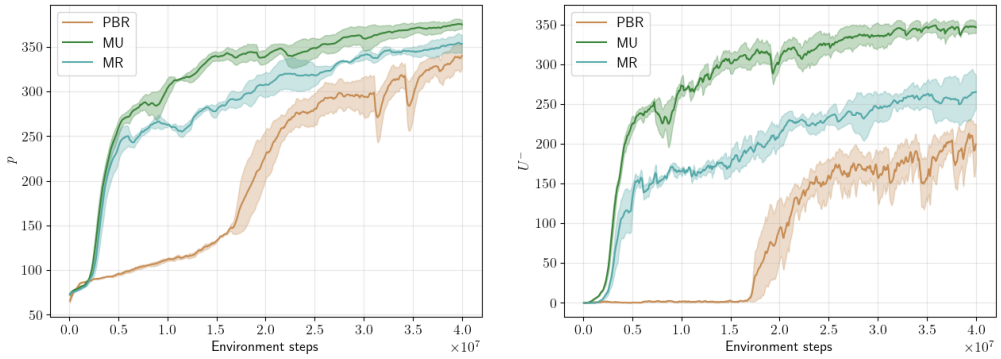
$$\beta_{t+1} = \mathcal{P}(\beta_t - \eta_\beta \nabla_\beta U(\pi_{\theta_t}, \beta_t)) \tag{9}$$

solves the maximin utility problem instead.

B Additional experimental results

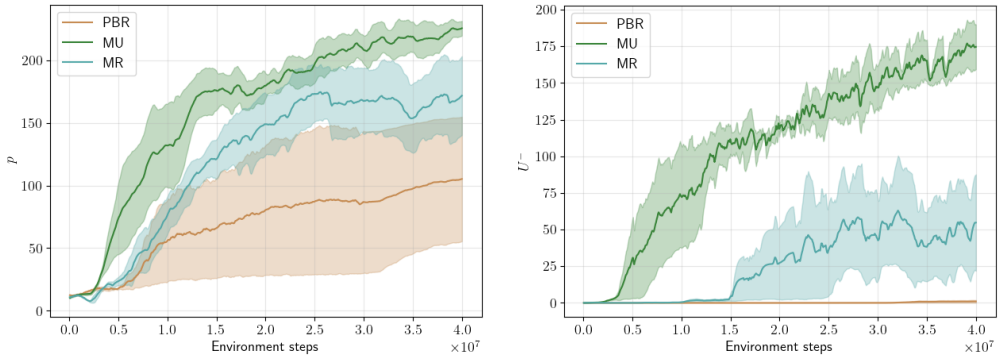
B.1 Robust Cooperation in Deep RL Tasks

We provide learning curves for the Collaborative Cooking game. For both kitchen layouts, the curves in Figures 2 and 3 uncover the fact that both the minimax regret and maximin utility formulations significantly speed up learning. The prior curves provided in Figures 4 and 5, highly smoothed for interpretability, show how different distributions are learned with utility and regret.



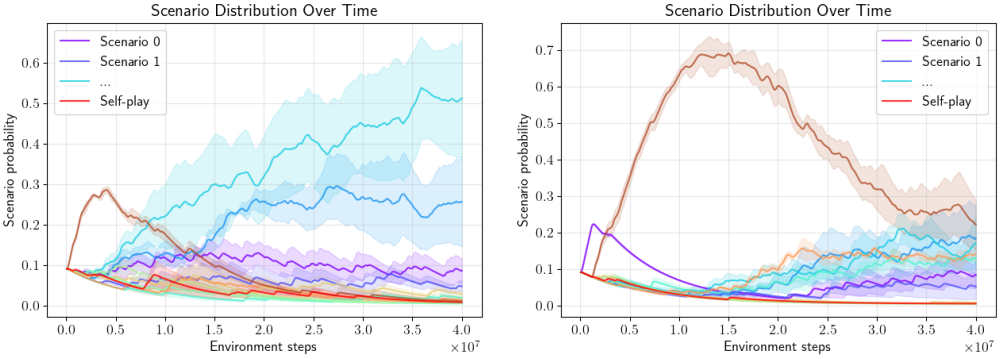
(a) Performance. (b) Worst-case utility.

Figure 2: Learning curves of the average and worst-case utility metrics over the training set of the Collaborative Cooking Cramped environment. The standard error is taken over three random seeds.



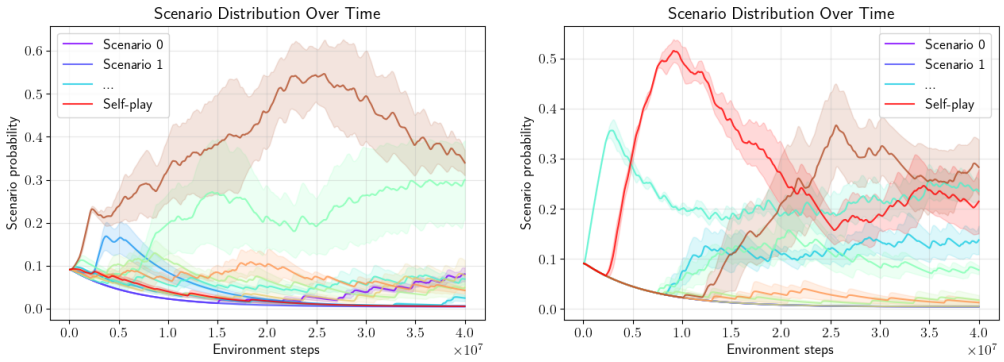
(a) Performance. (b) Worst-case utility.

Figure 3: Learning curves of the average and worst-case utility metrics over the training set of the Collaborative Cooking Circuit environment. The standard error is taken over three random seeds.



(a) Maximin Utility. (b) Minimax Regret.

Figure 4: Learning curves of the prior over the training scenarios, for the Collaborative Cooking Cramped environment. The standard error is taken over three random seeds.



(a) Maximin Utility.

(b) Minimax Regret.

Figure 5: Learning curves of the prior over the training scenarios, for the Collaborative Cooking Circuit environment. The standard error is taken over three random seeds.

C Additional experimental details

C.1 Robust Cooperation in Deep RL Tasks

To facilitate the training of our policies in Collaborative Cooking, we used a shaping pseudo-reward of 1 when tomatoes were placed in the cooking pot. For the background policies, we further altered the reward function to restrict delivery rewards to the player that delivered. Combining this new reward function with varying levels of prosociality and risk-aversion helped the background policies adopt diversified ways to solve the game.

The architecture for the agents consisted of a convolutional network with two layers, having 16 and 32 output channels, kernel shapes of 8 and 4, and strides of 8 and 1, respectively. The output of the convNet was concatenated with the previous action taken before being passed into a dense layer of size 256 and an LSTM with 256 units. Policy and baseline (for the critic) were produced by linear layers connected to the output of the LSTM.

We chose PPO to train our policies, using the Adam optimizer with a learning rate of 2×10^{-4}, a discount factor of 0.99, a GAE lambda of 0.95, a KL coefficient of 1.0 with a KL target of 0.01, and a PPO clipping parameter of 0.3. Gradients were clipped at 4.0. We did not employ entropy regularization. PPO was set to run 2 epochs per batch, each containing 64000 samples, with minibatches of 1000 samples each. Finally, the unroll length for the LSTM was set at 20.

For the prior, we used a learning rate of 0.4. We also constrained the prior to keep a probability of $5e-2$ to sample a random scenario in order to allow constant exploration.