A Minimax Approach to Ad Hoc Teamwork

Victor Villin
Université de Neuchâtel
Neuchâtel, Switzerland
victor.villin@unine.ch

Thomas Kleine Buening
The Alan Turing Institute
London, United Kingdom
tbuening@turing.ac.uk

Christos Dimitrakakis
Université de Neuchâtel
Neuchâtel, Switzerland
christos.dimitrakakis@unine.ch

ABSTRACT

We propose a minimax-Bayes approach to Ad Hoc Teamwork (AHT) that optimizes policies against an adversarial prior over partners, explicitly accounting for uncertainty about partners at time of deployment. Unlike existing methods that assume a specific distribution over partners, our approach improves worst-case performance guarantees. Extensive experiments, including evaluations on coordinated cooking tasks from the Melting Pot suite, show our method's superior robustness compared to self-play, fictitious play, and best response learning. Our work highlights the importance of selecting an appropriate training distribution over teammates to achieve robustness in AHT.

KEYWORDS

Multi-Agent Reinforcement Learning; Ad Hoc Teamwork

ACM Reference Format:
Victor Villin, Thomas Kleine Buening, and Christos Dimitrakakis. 2025. A Minimax Approach to Ad Hoc Teamwork. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.

1 INTRODUCTION

Domain generalisation is often crucial in Reinforcement Learning (RL) and is typically assessed by placing an agent in novel environments [13]. Likewise, in Multi-Agent Reinforcement Learning (MARL), generalisation to new agents can be evaluated by pairing a trained policy with unseen actors [1, 5, 21, 26]. While zero-shot domain adaptation is a valuable property [20, 40], it is equally important to ensure proper transfer to new behaviours in multi-agent settings, especially in situations where undesired interactions may arise [17]. More specifically, Ad Hoc Teamwork (AHT) occurs when an agent, initially unfamiliar with its teammates, 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 [23, 43].

Efforts in AHT have primarily focused on learning and inferring behavioural models or teammates types [2, 3, 5, 12, 33], adapting to behaviour shifts [37], and enhancing generalisation by encouraging diversity in partners during training [11, 21, 22, 30, 44]. However, these methods provide limited guarantees on the 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 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 different habits and expertise. In each of these scenarios, we can adopt the perspective that the *focal* actors are controlled by an automated agent, whereas the other actors are viewed as fixed, forming the *background* of the task [1, 26]. These scenarios can be viewed as distinct 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 [30, 44]. However, this only optimises performance for that specific distribution.

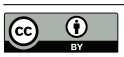
Recent studies in zero-shot domain transfer showed that selecting an appropriate prior over training environments is key to learning robust policies [8, 14, 16, 24, 27, 36]. Intuitively, this insight should apply to the AHT setting as well, suggesting that choosing a specific prior over scenarios/partners 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 training set of partners given our policy, an idea adopted from the minimax-Bayes concept [6].

Contributions. We make the following contributions:

(1) We adapt Minimax-Bayes Reinforcement Learning (MBRL) [8] 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 for AHT robustness, and provide solutions to target either metric (Section 5).
(3) We study out-of-distribution robustness guarantees (Section 6).
(4) We propose a Gradient Descent-Ascent (GDA) [29] based algorithm, in conjunction with policy-gradient methods, and discuss its convergence for softmax policies (Section 7).
(5) We conduct extensive experiments to evaluate our approach. We deploy learned policies on both seen and unseen scenarios for cooperative problems, including a partially observable cooking task from the Melting Pot suite [1, 26]. We compare our approach against Self-Play (SP), Fictitious Play (FP) [7, 19] as well as learning a policy w.r.t. a fixed uniform distribution over scenarios [30], which is related to fictitious co-play [44], as both learn the best response to a population of policies (Section 8).
(6) 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.

2 RELATED WORK

Ad Hoc Teamwork. In AHT, we are interested in developing agents capable of cooperating with other unfamiliar agents without



Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

any form of prior coordination [3, 5, 39, 43]. Popular approaches usually involve some form of Population Play (PP), where policies forming a population are learning by interacting with each other [1, 26, 30, 32]. Key strategies for ensuring generalisation to new partners include promoting policy diversity within the training population [11] and preventing overfitting to training partners [25]. Both Lupu et al. [30] and Strouse et al. [44] previously showed that learning a best response to a more diverse population leads to improved generalisation. Additionally, Jaderberg et al. [22] 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 [1, 10, 26].

To further improve AHT, several works suggest inferring the teammates' models/types, maintaining a belief about ad hoc partners based on previous interactions within an episode [2, 5]. This was shown to help substantially in the partially observable setting [15, 18, 38]. Efforts have also been made to improve the learning and generalisation of such models to new partners [3, 4, 33].

An alternative approach proposed by Li et al. [28] 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, whereas our approach is compatible with any policy-gradient algorithm.

Even though the aforementioned methods attempt to improve cooperative robustness, they always assume specific distributions for the partners. For example, Jaderberg et al. [22] used a distribution favoring the matchmaking of policies of similar levels with the intuition that the reward signal is stronger in those cases. However, it does not provide any insights on its effects on AHT robustness. As a result, the actual impact of the training partner distribution on robustness is left under-explored. This holds significant potential, as it can be exploited in conjunction with previously studied mechanisms to substantially enhance AHT robustness.

Zero-shot Domain Transfer. AHT can be seen as a form of zero-shot domain transfer. Each possible team composition involving the focal agent can be considered a different environment. In the single-agent setting, Jiang et al. [24] 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. [16] 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 tampering with the environment distribution was also explored by Pinto et al. [36], who employed a maximin utility formulation to choose continuous adversarial environment perturbations throughout learning. Instead of utility, Dennis et al. [14] stressed the advantages of using regret by proposing a training environment sampling scheme avoiding entirely unsolvable and uninformative environments. Most relevant to this work, Buening et al. [8] conducted a study over worst-case priors (for both utility and regret) over training environments, and proved that worst-case distributions are well-suited for domain transfer. Li et al. [27] later reaffirmed those results, learning worst-case distributions within ambiguity sets of subjective priors. Finally, there exist works on domain transfer in the MARL setting [40], 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 $\mu = \langle \mathcal{S}, \mathcal{X}, \mathcal{A}, O, P, \rho, T \rangle$ defined on finite sets of states $\mathcal{S}$, observations $\mathcal{X}$ and actions $\mathcal{A}$. The observation function $O : \mathcal{S} \times \{1, \dots, m\} \to \mathcal{X}$ 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. We denote $\|\rho\|_\infty$ the maximum absolute step reward.

A policy $\pi : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \times \mathcal{A} \times \cdots \times \mathcal{X} \to \Delta(\mathcal{A})$ 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* $\sigma = (c, \boldsymbol{\pi}^b)$ be defined by its number of *focal* players c, and its *background* players $\boldsymbol{\pi}^b = (\pi^b_1, \dots, \pi^b_{m-c}) \in \Pi^{m-c}$. We say we deploy a policy π^f in scenario σ if the c focal players are equal to 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 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:[1]

$$\Sigma(\mathcal{B}) \coloneqq \{(c, \boldsymbol{\pi}^b) \mid 1 \le c \le m, \boldsymbol{\pi}^b \in \mathcal{B}^{m-c}\}. \tag{1}$$

A scenario σ on μ can be viewed as its own c-player POMG, through the marginalisation of the policies of its background players.[2] We denote $\mu(\sigma) = \langle \mathcal{S}, \mathcal{X}, \mathcal{A}, O_\sigma, P_\sigma, \rho_\sigma, \gamma, T \rangle$ the POMG induced by scenario σ, where $O_\sigma : \mathcal{S} \times \{1, \dots, c\} \to \mathcal{X}$ is the corresponding observation function, $P_\sigma : \mathcal{S} \times \mathcal{A}^c \to \Delta(\mathcal{S})$ is the transition function given by

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

[1] The definition of background populations in (1) is largely inspired from the work of Leibo et al. [26] and Agapiou et al. [1]. This is the most general formulation of the problem and will be used as is throughout this work. Depending on the game μ, a restricted set may make more sense, such as limiting to $c = 1$.

[2] Each scenario can be seen as a decentralised partially observable Markov decision process [34] constrained by the fact that the c players are copies.

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

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

where h_j is the history of observations and actions of the j-th policy and $\mathbf{a}_j^b$ its action in $\mathbf{a}^b$. We denote the scenario that only involves the focal policy, i.e. the universalisation scenario [26], by $\sigma^{\text{SP}} = (m, \emptyset)$. This setup is closely related to N-agent AHT [46], in that the proportion of focal/background players can vary and is not known in advance.

3.3 Evaluation

The expected utility of a policy π in scenario σ is the mean return of the focal policies given by the expected *focal-per-capita return* [1, 26]:

$$U(\pi, \sigma) \coloneqq \sum_{t=1}^{T} \frac{1}{c} \sum_{i=1}^{c} \mathbb{E}^{\pi}_{\mu(\sigma)} \left[\rho_\sigma(s_t, \mathbf{a}_t^f, i) \right]. \tag{2}$$

$U^*(\sigma) \coloneqq \max_{\pi \in \Pi} U(\pi, \sigma)$ 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:

$$R(\pi, \sigma) \coloneqq U^*(\sigma) - U(\pi, \sigma). \tag{3}$$

To assess a learning method in terms of AHT, we use the evaluation protocol of Leibo et al. [26]. This has two phases:

(1) **Training phase**: A background population $\mathcal{B}^{\text{test}}$ is kept hidden. The policy learner has access to the game μ with no restrictions, apart from accessing $\mathcal{B}^{\text{test}}$. For example, the learner is free to use a modified instance μ' of μ, where the observation function, O, may be adjusted to include information about other players, or where the reward function, ρ, could be altered to provide joint rewards instead of individual ones.
(2) **Testing phase**: The obtained policy is fixed 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 unseen test scenarios $\Sigma^{\text{test}} \subset \Sigma(\mathcal{B}^{\text{test}})$:

$$U_{\text{avg}}(\pi, \Sigma) \coloneqq \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} U(\pi, \sigma), \tag{4}$$

In addition, we consider two metrics related to robustness, namely worst-case utility and worst-case regret:

$$U_{\min}(\pi, \Sigma) \coloneqq \min_{\sigma \in \Sigma} U(\pi, \sigma), \qquad R_{\max}(\pi, \Sigma) \coloneqq \max_{\sigma \in \Sigma} R(\pi, \sigma). \tag{5}$$

Maximising $U_{\min}$ 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 $R_{\max}$ avoids decisions that lead to significantly worse utility than the optimal utility.

The final objective is to design a learning process outputting a policy that reliably maximises utility or minimises regret on possibly unseen scenarios.

3.4 General Assumptions

To ensure our setting aligns with the AHT literature, we must adhere to three assumptions [31]: 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. b) There is no control over teammates, the learner can control its own copies but not other agents in the configuration. c) All agents (focal and background) are assumed to have a partially shared objective. Their reward function may be slightly 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.

4 ACHIEVING ROBUST AHT

To learn a policy able to cooperate with new partners, a straightforward idea is to reconstruct scenarios that would 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 connections of our setting to Minimax-Bayes Reinforcement Learning (MBRL).

4.1 Constructing Training Scenarios

Before learning any robust policy, we need to construct a diverse set of scenarios. A background population that encompasses a wide range of behaviours is needed in order to reconstruct scenarios existing in nature. Previous work on AHT tackled the issue in various manners, such as using genetic algorithms [33], rule-based policies generated with MAP Elites [9], SP policies [44], explicit behavior diversification through regularisation [30], or through evolved pseudo-rewards [22]. Based on real-life examples and aiming to thoroughly assess the effects of partner priors, we adopt the following approach:

- We assume a class of reward functions for background policies:

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

  with ρ_{social} defined as

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

  where ρ^+ and ρ^- are the positive and negative parts of ρ, and λ_i and δ_i denoting levels of prosociality [35] and risk-aversion, respectively. In other words, each background policy has their own preferences (λ_k, δ_k).
- Policies are organised into sub-populations $\mathcal{B} = \bigcup_k \mathcal{B}_k$ of varying sizes.
- Each sub-populations are separately trained using PP.

Given the diverse preferences and varying sizes of the sub-populations, distinct habits and established conventions are more likely to emerge from each group [44]. This choice for constructing scenarios ensures a diverse generation of scenarios, important to ablate the

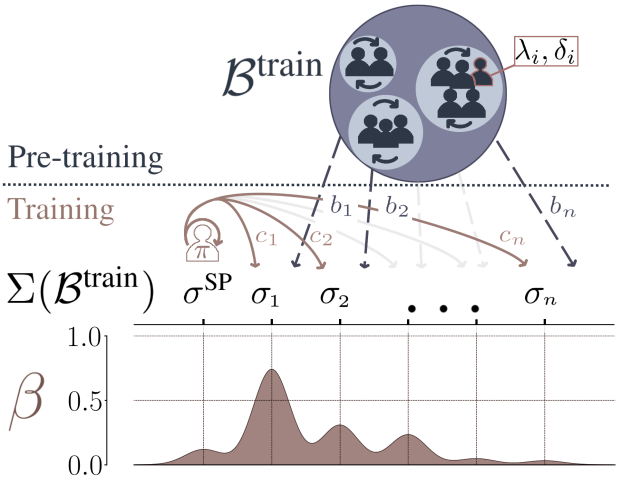


Figure 1: Illustration of the framework used in this paper. Prior to training the focal policy π, background policies with different preferences (λ_i, δ_i) learn by interacting within sub-populations of varying sizes. These sub-populations are then combined to form a background population, $\mathcal{B}^{\text{train}}$, used as a common 'training dataset' for all algorithms.
Our primary focus is on the training phase, where the focal policy π is trained while the distribution β over scenarios is tuned according to the proposed minimax game. 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.

effects of various scenario priors on AHT robustness. Note that this choice for constructing scenarios remains arbitrary and is not the main focus of our work.

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. [8] 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 policy more robust, as it prepares for the worst outcomes.

To transfer this idea to our setting, we remark that any finite background population $\mathcal{B}$ provides a finite set of POMGs $\mathcal{M}_{\mathcal{B}} = \{\mu(\sigma) | \sigma \in \Sigma(\mathcal{B})\}$. The principal 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 $\beta \in \Delta(\Sigma(\mathcal{B}))$:

$$U(\pi, \beta) \coloneqq \mathbb{E}_{\sigma\sim\beta}[U(\pi, \sigma)] = \sum_{\sigma} U(\pi, \sigma)\beta(\sigma). \tag{6}$$

This allows us to formulate the following maximin game:[3]

$$\max_{\pi\in\Pi} \min_{\beta\in\Delta(\Sigma(\mathcal{B}))} U(\pi, \beta). \tag{7}$$

Similarly to Buening et al. [8], 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 nature's distribution does not depend on the policy that will be picked. Fortunately, (7) has a value when $\mathcal{B}$ is finite.

Corollary 4.1 (Buening et al. [8]). *For an m-player POMG μ in a finite state-action space, with a known reward function and a finite horizon, and a background population $\mathcal{B}$, the maximin game* (7) *has a value:*

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

Proof. First, observe that for any stochastic policy $\pi \in \Pi$, there exists a distribution over deterministic policies $\phi \in \Delta(\Pi^{\text{D}})$ such that $\pi(a_t|h_t) = \sum_{d\in\Pi^{\text{D}}} d(a_t|h_t)\phi(d)$. Consequently, we can rewrite the utility as $U(\pi, \beta) = \sum_{d\in\Pi^{\text{D}}} \sum_{\sigma\in\Sigma(\mathcal{B})} U(d, \sigma)\phi(d)\beta(\sigma)$. 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 arbitrarily a 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 (7) has its utility lower-bounded on $\Sigma(\mathcal{B})$:

$$\forall \beta \in \Delta(\Sigma(\mathcal{B})), \quad U(\pi^*_U, \beta) \geq U(\pi^*_U, \beta^*_U), \tag{9}$$

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:

$$\forall \pi \in \Pi, \quad U(\pi^*_U, \beta^*_U) \geq U(\pi, \beta^*_U). \tag{10}$$

Note that this differs fundamentally from merely finding the best response to a fixed worst-case prior $\arg\max_\pi U(\pi, \beta^*_U)$, which once again, only has a guaranteed optimal utility on β^*_U.

Corollary 4.2 (Buening et al. [8]). *For any policy $\pi \in \Pi$ and background population $\mathcal{B} \subset \Pi$, we have*

$$\min_{\beta\in\Delta(\Sigma(\mathcal{B}))} U(\pi, \beta) = U_{min}(\pi, \Sigma(\mathcal{B})). \tag{11}$$

Proof. This follows directly from the results of Buening et al. [8], using utility in place of regret and recognising that Dirac distributions associated with scenarios in $\Sigma(\mathcal{B})$ are always contained in $\Delta(\Sigma(\mathcal{B}))$. □

Lemma 4.3. *For any background population $\mathcal{B} \subset \Pi$ and π^*_U the policy solving the maximin utility game* (7)*, we have*

$$U_{min}(\pi^*_U, \Sigma(\mathcal{B})) = \max_{\pi\in\Pi} U_{min}(\pi, \Sigma(\mathcal{B})). \tag{12}$$

Proof. By Corollary 4.2, we can write that $\max_\pi \min_\beta U(\pi, \beta) = \max_\pi U_{\min}(\pi, \Sigma(\mathcal{B}))$. However, we also have $\max_\pi \min_\beta U(\pi, \beta) = \min_\beta U(\pi^*_U, \beta) = U_{\min}(\pi^*_U, \Sigma(\mathcal{B}))$. □

[3] If we have a subjective prior, we could learn the distribution within an ϵ-ball around that prior [27]. We however consider the full simplex for simplicity.

Thus, a policy solving the maximin utility game (7) is guaranteed to have an optimal worst-case utility on its training set.

5 UTILITY OR REGRET?

Optimising for the worst-case utility (7) might be problematic. Nature could resort to only picking scenarios where the focal players achieve the worst possible score. Then, the distribution trivially minimises utility for any chosen policy, preventing the latter to learn anything. Buening et al. [8] 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 $L(\pi, \beta) \coloneqq \sum_\sigma R(\pi, \mu)\beta(\sigma)$ be the Bayesian regret with respect to a prior β, we now formulate the following minimax regret game:

$$\min_{\pi \in \Pi} \max_{\beta \in \Delta(\Sigma(\mathcal{B}))} L(\pi, \beta). \tag{13}$$

One can also prove that this above game has a value. Moreover, a solution (π_R^*, β_R^*) to (13) exhibits properties analogous to those in equations (9), (10) and (12), but in terms of regret. π_R^* has its Bayesian regret upper-bounded by $L(\pi_R^*, \beta_R^*)$ on $\Sigma(\mathcal{B})$. It is also optimal under the worst-case prior β_R^* and achieves optimal worst-case regret $R_{\max}$ on $\Sigma(\mathcal{B})$.

Should utility or regret be used as an objective? Exploiting regret ensures that scenarios on which you can improve the most 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 5.1 (Non-degenerative population). A background population of policies $\mathcal{B} \subset \Pi$ is non-degenerative if and only if for any scenario $\sigma \in \Sigma(\mathcal{B})$, there exists two distinct policies π_1 and $\pi_2 \in \Pi, \pi_1 \neq \pi_2$ such that $U(\pi_1, \sigma) \neq U(\pi_2, \sigma)$.

Lemma 5.2. *If a background population $\mathcal{B} \subset \Pi$ is non-degenerative, then for any scenario $\sigma \in \Sigma(\mathcal{B})$, there exists a policy $\pi \in \Pi$ such that $R(\pi, \sigma) > 0$.*

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 background population, no distribution can deadlock the policy learner into stale scenarios. Hence, the utility-minimising opponent in Equation 7 can no longer trivially minimise utility. For the remainder of the paper, background populations are assumed to be non-degenerative.

6 OUT-OF-DISTRIBUTION ROBUSTNESS

As already stated in Section 4.1, having a diverse set of scenarios that adequately represents the true set of scenarios is crucial. However, since it is often impractical to perfectly replicate the true set, the prior used during training may not have the same support as the true distribution observed in nature. In such cases, the guarantees outlined in Section 4.2 no longer hold on the true distribution. In order to state further robustness guarantees, an option is to assume that scenarios in the true scenario set are close to the training scenarios. To quantify the closeness between scenarios, we first define the distance between two policy vectors as their maximum total variation across all states:

$$d(\boldsymbol{\pi}, \boldsymbol{\pi}') = \max_{s \in \mathcal{S}} \sum_i \sum_a \left| \boldsymbol{\pi}_i(a|s) - \boldsymbol{\pi}'_i(a|s) \right|. \tag{14}$$

We define the scenario distance as the minimum distance between policy vectors across permutations of the background policies:

$$d(\sigma, \sigma') = \min_{\boldsymbol{\pi}, \boldsymbol{\pi}' \in \mathrm{Perm}(\boldsymbol{\pi}^b) \times \mathrm{Perm}(\boldsymbol{\pi}^{b'})} d(\boldsymbol{\pi}, \boldsymbol{\pi}'), \tag{15}$$

This metric measures the similarity between the background policies of two scenarios. Scenarios can only be compared if they have the same number of focal players (e.g., $\sigma = (c, \boldsymbol{\pi}^b)$ and $\sigma' = (c, \boldsymbol{\pi}^{b'})$).

Definition 6.1 (ϵ-net of a scenario set). A finite set of scenarios Σ is called an ϵ-net of a scenario set S if and only if, for every scenario $\sigma \in S$, there exists a scenario $\sigma' \in \Sigma$ such that $d(\sigma, \sigma') < \epsilon$.

Lemma 6.2. *Let Σ be an ϵ-net for a scenario set S. For any policy $\pi \in \Pi$ and scenario $\sigma \in S$, there is a scenario $\sigma' \in \Sigma$ that verifies:*

$$\left| U(\pi, \sigma) - U(\pi, \sigma') \right| < \frac{\epsilon T^2 \|\rho\|_\infty}{2}. \tag{16}$$

Proof Sketch. The result is obtained by using the fact that for any pairs of ϵ-close scenarios σ, σ' and any $s, \mathbf{a}^f, i$, we have $\sum_{s'} |P_\sigma(s'|s, \mathbf{a}^f) - P_{\sigma'}(s'|s, \mathbf{a}^f)| < \epsilon$ and $|\rho_\sigma(s, \mathbf{a}^f, i) - \rho_{\sigma'}(s, \mathbf{a}^f, i)| < \epsilon \|\rho\|_\infty$. The proof is concluded by showing by induction that for all t and s , $|U_t(\pi, \sigma, s) - U_t(\pi, \sigma', s)| < \frac{1}{2}\epsilon(T-t+1)(T-t)\|\rho\|_\infty$. □

Lemma 6.3. *Let Σ be an ϵ-net for some scenario set S. For any policy $\pi \in \Pi$ and scenario $\sigma \in S$, there is a scenario $\sigma' \in \Sigma$ such that*

$$\left| R(\pi, \sigma) - R(\pi, \sigma') \right| < \epsilon T^2 \|\rho\|_\infty. \tag{17}$$

Proof Sketch. The result is obtained by both using the identity $|U^*(\sigma) - U^*(\sigma')| \leq \max_\pi |U(\pi, \sigma) - U(\pi, \sigma')|$ and noticing that for any policy π, $|R(\pi, \sigma) - R(\pi, \sigma')| \leq |U^*(\sigma) - U^*(\sigma')| + |U(\pi, \sigma) - U(\pi, \sigma')|$. □

Lemma 6.4. *Let Σ be an ϵ-net for some scenario set S, and π_U^* the optimal policy for the maximin utility problem (7) on Σ, then*

$$U_{min}(\pi_U^*, S) > \max_{\pi \in \Pi} \left(U_{min}(\pi, \Sigma) - \frac{\epsilon T^2 \|\rho\|_\infty}{2} \right). \tag{18}$$

Proof Sketch. We denote $\sigma_{\mathrm{wc}}(\Sigma)$ and $\sigma_{\mathrm{wc}}(S)$ the worst-case scenarios for π_U^* on Σ and S, and reason on the distance between $\sigma_{\mathrm{wc}}(\Sigma)$ and $\sigma_{\mathrm{wc}}(S)$. If $d(\sigma_{\mathrm{wc}}(\Sigma), \sigma_{\mathrm{wc}}(S)') < \epsilon$, then Lemma 6.2 applies. Otherwise, since Σ is an ϵ-net, we can find another scenario $\sigma_\epsilon \in \Sigma$ that is ϵ-close to $\sigma_{\mathrm{wc}}(S)$ and use the fact that the utility of π_U^* is by definition higher on σ_ϵ than on $\sigma_{\mathrm{wc}}(\Sigma)$. □

Algorithm 1 Background-Focal GDA

Input Background policies $\mathcal{B}$, and learning rates (η_π, η_β).
Simplex projector $\mathcal{P}$
Initialise the main policy parameters θ_0 randomly
Initialise the distribution as uniform $\beta_0 = \mathcal{U}(\Sigma(\mathcal{B}))$
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 $L(\pi_{\theta_t}, \beta_t) = \sum_\sigma R(\pi_{\theta_t}, \sigma)\beta_t(\sigma)$
  if solving maximin utility (7) **then**
    Update distribution $\beta_{t+1} = \mathcal{P}(\beta_t - \eta_\beta \nabla_\beta U(\pi_{\theta_t}, \beta_t))$
  if solving minimax regret (13) **then**
    Update distribution $\beta_{t+1} = \mathcal{P}(\beta_t + \eta_\beta \nabla_\beta L(\pi_{\theta_t}, \beta_t))$
  Update policy parameters $\theta_{t+1} = \theta_t + \eta_\theta \nabla_\theta U(\pi_{\theta_t}, \beta_t)$
return θ^*, β^* uniformly sampled from $\{(\theta_1, \beta_1), \dots, (\theta_N, \beta_N)\}$

Lemma 6.5. *Let Σ be an ϵ-net for some scenario set S, and π_R^* the optimal policy for the minimax regret problem* (13) *on Σ, then*

$$R_{max}(\pi_R^*, S) < \min_{\pi \in \Pi} \left(R_{max}(\pi, \Sigma) + \epsilon T^2 \|\rho\|_\infty \right). \tag{19}$$

Proof Sketch. We prove, analogically to Lemma 6.4, the result using Lemma 6.3 in place of Lemma 6.2. □

Lemmas 6.4 and 6.5 provide worst-case guarantees on arbitrary sets of scenarios, for policies solving the minimax problems. This also means that we can have those guarantees on non-finite sets of scenarios. Importantly, as long as we have an ϵ-net of training scenarios for the true set, the policy solving the maximin utility (or minimax regret) problem has a strong worst-case utility (or regret) guarantee. In contrast, it is impossible to guarantee *anything* additional about the average utility U_{avg} on the true set, as the latter could very well include scenarios that are all ϵ-close to the worst-case scenarios of the training set. For this reason, the average utility on the true set can be as low as the worst-case utility.

7 COMPUTING SOLUTIONS

We now desire to calculate the solution pairs for both the maximin utility (7) and minimax regret (13) games. Buening et al. [8] theoretically proved that GDA has convergence guarantees when the game is played between a policy learned with softmax parameterisation 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 favours proper convergence by treating the copies as fixed policies.

Algorithm 1, a GDA algorithm adapted to our setting, can be employed to learn solutions for both the maximin utility and minimax regret problems. Furthermore, in case the POMG is not known, one can straightforwardly adapt the algorithm to a stochastic version, by resorting to sampling scenarios and performing policy rollouts to estimate utility, regret, and gradients.

Table 1: Payoff matrix of the Prisoner's Dilemma.

	Cooperate	Defect
Cooperate	(4, 4)	(0, 5)
Defect	(5, 0)	(1, 1)

8 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 Iterated Prisoner's Dilemma to validate the theoretical results. Following this, we test our approaches on a deep-RL task, the Collaborative Cooking (Overcooked) game [1, 10, 26, 44]. We remind that we want to optimise policies with respect to the focal-per-capita return (2) rather than individual returns. Naturally, across all of our experiments, we reward focal policies with average focal step rewards. For each scenario σ, this is defined as $\rho_\sigma^{\text{train}}(s, \mathbf{a}^f, i) = \frac{1}{c} \sum_{j=1}^{c} \rho_\sigma(s, \mathbf{a}^f, j)$. Throughout our experiments, we benchmark against three distribution management strategies:

- Population Best Response (PBR): uniform distribution over scenarios,
- Self-Play (SP): playing solely with copies of the focal policy,
- Fictitious Play (FP): sampling partners uniformly from the history of focal policies $\pi_0, \dots, \pi_t$.

8.1 Iterated 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 Iterated Prisoner's Dilemma, where two players play the matrix game (Table 1) repeatedly for $T = 3$ rounds.

Experimental Setup. In the Iterated Prisoner's Dilemma, players receive and observe rewards based on their chosen actions, specified by the payoff matrix. The game has one state, and the outcomes observed are enough to determine the joint actions, making it fully observable. We learn softmax, fully adaptive policies, where actions depend on the entire history of observations and actions. Unlike other experiments, we do not use the approach described in Section 4.1. Instead, the learner interacts with a background population $\mathcal{B}^{\text{train}}$ composed of nine ad-hoc policies, such as pure cooperation/defection, tit-for-tat, cooperate-until-defected, and fully random. For AHT assessment, we adequately construct a test background population $\mathcal{B}^{\text{test}}$, ensuring that its scenarios are ϵ-close to those in the training phase. Specifically, we uniformly sample 512 stochastic policies that are ϵ-close to the policies in the training background population, setting $\epsilon = 0.5$.

Results. Table 2 summarises the performance on both training and test sets. As expected, on the training set, PBR performs the best under the uniform prior (U_{avg}), MU has the highest worst-case utility (U_{min}), and MR exhibits the lowest worst-case regret (R_{max}).

Table 2: Scores on the Iterated Prisoner's Dilemma. A higher value is desired for average utility (U_{avg}) and worst-case utility (U_{min}), while a lower worst-case regret (R_{max}) is better.

	$\Sigma(\mathcal{B}^{\text{train}})$			$\Sigma(\mathcal{B}^{\text{test}})$		
	U_{avg}	U_{min}	R_{max}	U_{avg}	U_{min}	R_{max}
Maximin Utility (MU)	7.69	**3.00**	9.00	**8.34**	**3.00**	9.00
Minimax Regret (MR)	8.23	2.26	**3.79**	7.96	2.78	**4.35**
Population Best Response (PBR)	**8.54**	2.00	4.97	8.07	2.48	5.63
Fictitious Play (FP)	7.06	0.14	10.56	6.33	0.56	9.96
Self-Play (SP)	7.25	0.47	10.19	6.49	0.86	9.65
Random	7.40	1.50	5.50	7.47	2.18	5.20

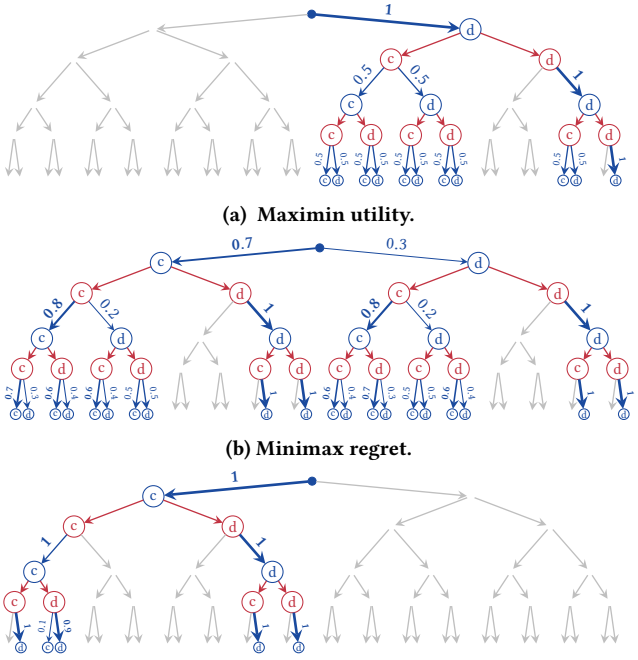
On the test set however, MU outperforms PBR in terms of average utility and maintains the highest worst-case utility, while MR continues to excel in minimising worst-case regret. These results indicate that best responses to populations does not ensure the best robustness to new partners. SP and FP agents as well, overfit to their established conventions, leading to poor transferability across training and test policies. Figure 2 visualises the learned policies under different distribution regimes, highlighting their disparity. For example, the population best response is a strategy close to "cooperate-until-defected", while MU's policy heavily favors defection. Crucially, this figure points to a potential improvement for future work: during optimisation, the worst-case distribution can force policies onto a narrow subset of scenarios, leaving others unexplored. Due to the MU regime, the policy is forced to face a pure defecting opponent, its worst-case scenario, entirely cutting its exposure to cooperative strategies. As seen in Figure 2a, the policy does not know what to do if the opponent chooses to collaborate. This suggests that it is possible to have a policy with equal worst-case utility but improved worst-case regret and overall performance.

8.2 Robust AHT on Collaborative Cooking

For this section, we tackle the Collaborative Cooking game [1], 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. Each have to collect tomatoes, cook them, prepare dishes, and deliver the soup. Successful deliveries reward both players equally.[4] Players must navigate the kitchen, interact with objects in the right order, and coordinate with each other. Each player has a local, 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 for all background policies are sampled uniformly in $[-0.2, 1.2]$ and $[0.1, 2]$ respectively.[5] For a fair comparison and to focus on scenario distribution learning, we assume that $\mathcal{B}^{\text{train}}$ is readily available to all approaches,

[4]When training background policies specifically, we only assign a reward of 20 points to the player who delivers the dish, and 1 point to players who contribute by placing a tomato into the pot. This incentivises diverse behaviour generation.
[5]Those intervals were chosen empirically to ensure diverse and cooperative policies.

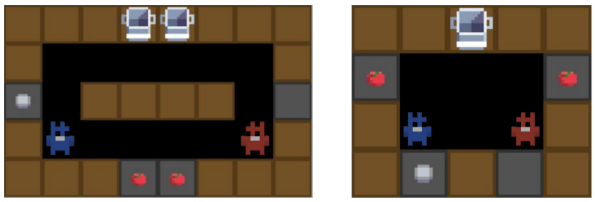


(a) Maximin utility.

(b) Minimax regret.

(c) Population best response.

Figure 2: The different policies obtained in function of their training distribution regime. "c" and "d" stands for the "cooperate" and "defect" actions, respectively. The learned policy action probabilities (rounded to one decimal) are in blue, while the opponent actions are in red. Players take actions simultaneously over the course of three rounds (opponent's last action is not shown).



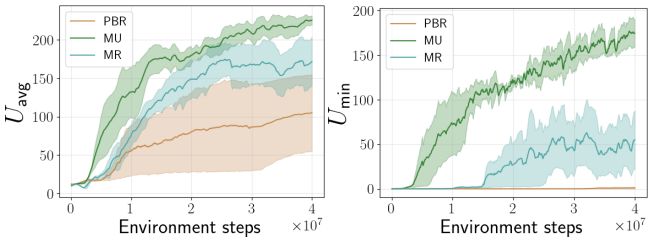
(a) Circuit. (b) Cramped.

Figure 3: We consider two kitchen layouts in the Collaborative Cooking environment. The players must collaborate to deliver tomato soups without blocking each other. Both players collect a reward of 20 each time a soup is delivered.

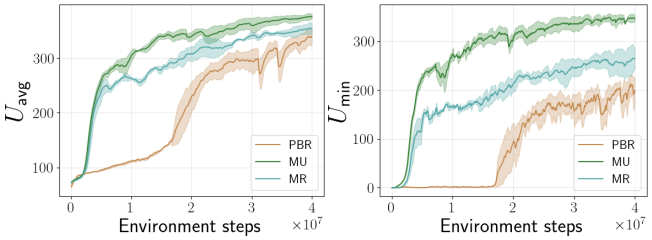
which can be exploited for a maximum of 4×10^7 environment steps to learn a policy with PPO [41]. We run each training procedure on three different seeds (solely $\mathcal{B}^{\text{train}}$ and $\mathcal{B}^{\text{test}}$ remain consistent across runs). We evaluate the approaches on two different kitchen layouts: Circuit and Cramped [1], which are visualised in Figure 3. On top of our own test scenarios, we assess the learned policies on the Melting Pot benchmark scenarios, comparing them against the baselines reported in the original paper [1]: an Actor-Critic

Table 3: Scores on the Collaborative Cooking environment training and test sets. The standard error is taken over three random seeds. The scores are aggregated over the two kitchen layouts. Lower worst-case regret $R_{\max}$ is better. Scores in italic are reported from Agapiou et al. [1], which were not obtained on the exact same setting.

	$\Sigma(\mathcal{B}^{\text{train}})$			$\Sigma(\mathcal{B}^{\text{test}})$			Melting pot scenarios		
	U_{avg}	$U_{\min}$	$R_{\max}$	U_{avg}	$U_{\min}$	$R_{\max}$	U_{avg}	$U_{\min}$	$R_{\max}$
Maximin Utility (MU)	**266.9 ± 4.3**	**225.3 ± 11.5**	266.0 ± 7.9	**195.7 ± 6.2**	**66.0 ± 6.8**	266.4 ± 10.1	**273.8 ± 4.9**	**224.9 ± 7.1**	**118.0 ± 7.1**
Minimax Regret (MR)	232.0 ± 18.6	144.3 ± 14.4	**230.7 ± 28.2**	172.2 ± 15.4	65.1 ± 16.0	**248.2 ± 28.4**	206.8 ± 12.6	148.7 ± 9.1	187.1 ± 13.0
Population Best Response (PBR)	209.7 ± 23.9	96.8 ± 13.4	357.6 ± 16.1	151.4 ± 19.5	33.6 ± 5.9	327.1 ± 14.0	171.8± 21.1	106.3 ± 9.3	228.1 ± 5.8
Fictitious Play (FP)	129.9 ± 13.9	0.2 ± 0.1	483.5 ± 16.1	152.2 ± 16.8	16.7 ± 11.7	369.7 ± 19.7	121.5 ± 15.7	40.7 ± 16.3	294.8 ± 10.7
Self-Play (SP)	124.8 ± 26.4	15.7 ± 10.5	460.8 ± 21.7	117.4 ± 12.5	6.7 ± 3.5	367.5 ± 11.6	101.2 ± 17.8	29.0 ± 17.4	293.4 ± 14.5
Random	42.8 ± 0.0	0.0 ± 0.0	505.4 ± 0.0	32.2 ± 0.0	0.0 ± 0.0	445.0 ± 0.0	60.6 ± 0.0	0.0 ± 0.0	307.3 ± 0.0
PP-ACB	n/a	n/a	n/a	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	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	n/a	n/a	n/a	*78.6 ± 0.0*	*36.1 ± 0.0*	*306.7 ± 0.0*



(a) Circuit.



(b) Cramped.

Figure 4: Learning curves of the average and worst-case utility metrics over the training set (averaged over 3 runs). The standard error is shown in shaded colour.

Baseline (ACB), V-MPO [42], and OPRE [45].[6] These baselines were trained for 10^9 steps without access to our background policies. In each scenario (including the Melting Pot scenarios), regret was computed by estimating best responses with PPO for 10^7 steps. Lastly for MU and MR, we run a stochastic version of Algorithm 1, and constrain the learned distribution to keep sampling a random scenario with probability 0.05, effectively allowing us to maintain utility estimates across all scenarios.

Results. The results in Table 3 clearly show that MU outperforms all other evaluated methods. Looking at the robustness metrics, the utility formulation has the best worst-case utility ($U_{\min}$) overall and the best worst-case regret ($R_{\max}$) 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 set and our own test set. In terms of average performance (U_{avg}), 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 possible explanation for the globally lower performance of the regret approach compared to utility is that the best responses for training scenarios are too approximate. Figure 4 suggests another hypothesis for why MU and MR considerably outperform leaning population best responses and other approaches: their scenario distributions during training have a similar effect to curriculum learning, introducing indirect exploration in behaviours compared to fixed distributions, and a smoother learning curve. In fact, they appear to speed up learning. In contrast, Figure 4a shows that training against a uniform distribution of scenarios fails to improve in at least one scenario on the Circuit layout, with a worst-case utility close to 0.

9 CONCLUSION

We explored how to compute robust adaptive policies for Ad Hoc Teamwork. Building on Minimax-Bayes RL, we introduced a method to identify minimax distributions over background populations, which consistently yielded more robust policies compared to training with a uniform distribution. Surprisingly, we also found that training on minimax distributions can significantly accelerate learning. However, utilising regret as an objective to tune the training distribution proved computationally expensive when best-response utilities are not readily available. In some special instances, we also found that the minimax-Bayes training approach w.r.t. utility can prevent policies from learning certain game dynamics.

Looking forward, we see great potential in extending our approach to a curriculum learning framework based on partner distributions, which could dramatically improve sample efficiency, asymptotic performance, and AHT robustness.

ACKNOWLEDGEMENTS

Thomas Kleine Buening is supported by the UKRI Prosperity Partnership Scheme (Project FAIR).

[6]Since their baseline policies are not publicly available, we were unable to evaluate them on our scenarios, which explains the missing values in Table 3.

REFERENCES

[1] 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]

[2] 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).

[3] 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.

[4] 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).

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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).

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Hasra Dodampegama and Mohan Sridharan. 2023. Knowledge-based reasoning and learning under partial observability in ad hoc teamwork. *Theory and Practice of Logic Programming* 23, 4 (2023), 696–714.

[16] 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).

[17] 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).

[18] Pengjie Gu, Mengchen Zhao, Jianye Hao, and Bo An. 2021. Online ad hoc teamwork under partial observability. In *International conference on learning representations*.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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).

[24] 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.

[25] 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.

[26] 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.

[27] Jack Li, Zeyu Dong, and Shuo Han. 2024. Bayes-Optimal, Robust, and Distributionally Robust Policies for Uncertain MDPs. (2024).

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] Darius Muglich, Christian Schroeder de Witt, Elise van der Pol, Shimon Whiteson, and Jakob Foerster. 2022. 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.

[33] Darius Muglich, Luisa M Zintgraf, Christian A Schroeder De Witt, Shimon Whiteson, and Jakob Foerster. 2022. 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.

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

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

[36] 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.

[37] 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.

[38] João G Ribeiro, Cassandro Martinho, Alberto Sardinha, and Francisco S Melo. 2023. Making Friends in the Dark: Ad Hoc Teamwork Under Partial Observability. In *ECAI 2023*. IOS Press, 1954–1961.

[39] 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.

[40] 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.

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

(2017).
[42] 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).
[43] 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.
[44] 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.
[45] 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.
[46] Caroline Wang, Arrasy Rahman, Ishan Durugkar, Elad Liebman, and Peter Stone. 2024. N-Agent Ad Hoc Teamwork. arXiv:2404.10740 [cs.AI]