Minimum Coverage Sets for Training Robust Ad Hoc Teamwork Agents

Muhammad Rahman[1], **Jiaxun Cui**[1], **Peter Stone**[1,2]
[1]Department of Computer Science, The University of Texas at Austin
[2]Sony AI
arrasy@cs.utexas.edu, cuijiaxun@utexas.edu, pstone@cs.utexas.edu

Abstract

Robustly cooperating with unseen agents and human partners presents significant challenges due to the diverse cooperative conventions these partners may adopt. Existing Ad Hoc Teamwork (AHT) methods address this challenge by training an agent with a population of diverse teammate policies obtained through maximizing specific diversity metrics. However, prior heuristic-based diversity metrics do not always maximize the agent's robustness in all cooperative problems. In this work, we first propose that maximizing an AHT agent's robustness requires it to emulate policies in the minimum coverage set (MCS), the set of best-response policies to any partner policies in the environment. We then introduce the L-BRDiv algorithm that generates a set of teammate policies that, when used for AHT training, encourage agents to emulate policies from the MCS. L-BRDiv works by solving a constrained optimization problem to jointly train teammate policies for AHT training and approximating AHT agent policies that are members of the MCS. We empirically demonstrate that L-BRDiv produces more robust AHT agents than state-of-the-art methods in a broader range of two-player cooperative problems without the need for extensive hyperparameter tuning for its objectives. Our study shows that L-BRDiv outperforms the baseline methods by prioritizing discovering distinct members of the MCS instead of repeatedly finding redundant policies.

1 Introduction

The *Ad Hoc Teamwork* (AHT) problem (Stone et al. 2010) is concerned with learning ways to quickly cooperate with previously unseen agents or humans (henceforth referred to as "*unseen*" or "*novel*" teammates, or when unambiguous, simply "teammates"). In problems with multiple ways to coordinate, agents co-trained with a limited set of teammates may settle on cooperation conventions that only work when they collaborate with each other. Specialization towards these conventions diminishes an agent's ability to collaborate with unseen partners that adopt other conventions (Hu et al. 2020).

Recent works address this problem by optimizing diversity metrics to generate sets of teammate policies for AHT training (Lupu et al. 2021; Strouse et al. 2021; Xing et al. 2021; Bakhtin et al. 2022). Through interaction with the generated broadly representative teammate policies, an agent learns a policy to interact with previously unseen partners based on limited interactions. State-of-the-art methods optimize adversarial diversity to generate *incompatible* teammate policies (Charakorn, Manoonpong, and Dilokthanakul 2023; Cui et al. 2023a; Rahman et al. 2023). They seek sets of teammate policies, each maximizing their returns when playing with a designated AHT agent policy while minimizing returns with other policies.

Such existing diversity metrics are heuristic in nature and are not well-justified. It is unclear whether and how optimizing them can lead to improved robustness in general cooperative problems. We further demonstrate that optimizing these diversity metrics can fail to discover teammate policies under certain conventions even in simple cooperative games, specifically if following a convention yields high returns against the best-response policy to another generated teammate policy. Optimizing adversarial diversity can also generate teammates adopting *self-sabotaging* policies (Cui et al. 2023a). Self-sabotage potentially increases the difficulty of AHT training since the generated teammate policies can undermine collaboration with the trained AHT agent.

In this work, we make three contributions that improve existing teammate generation methods for training robust AHT agents. First, we outline formal concepts describing an ideal set of teammate policies for training robust AHT agents, which can emulate the best-response policy to any teammate during interaction (Chakraborty and Stone 2014). The importance of finding the best-response policies to design a robust agent provides the motivation to estimate the **minimum coverage set (MCS)**, which is the set of best-response policies to any teammate policy in an environment, before interacting with unknown teammates. Second, we use the concept of MCS to propose the **L-BRDiv** algorithm[1] that jointly estimates the MCS of an environment and utilizes it to generate teammates for AHT training by solving a constrained optimization problem. L-BRDiv's generated set of teammate policies encourages AHT agents to emulate policies in the MCS through AHT training. Third, we provide experiments that empirically demonstrate that L-BRDiv produces more robust AHT agents than state-of-the-art teammate generation methods while requiring fewer hyperparameters to be tuned.

[1]Implementation of L-BRDiv is available at https://github.com/raharrasy/L-BRDiv. The appendix is also accessible through https://arxiv.org/abs/2308.09595.

2 Related Work

Ad Hoc Teamwork Assuming knowledge of teammate policies that will be encountered during evaluation, some existing AHT methods train adaptive AHT agents that can achieve near-optimal performance when interacting with any teammate policy encountered in evaluation (Mirsky et al. 2022). These methods equip an agent with two components. The first is a *teammate modeling component* that infers an unknown teammate's policy via observations gathered from limited interactions with the unknown teammate. The second is an *action selection component* that estimates the best-response policy to the inferred teammate policy, which selects actions that maximize the AHT agent's returns when collaborating with an unknown teammate. PLASTIC-Policy (Barrett et al. 2016) is an early example AHT method that defines an AHT agent policy based on the aforementioned components. Recent works (Rahman et al. 2021; Zintgraf et al. 2021; Papoudakis, Christianos, and Albrecht 2021; Gu et al. 2021) implement these two components as neural network models which are trained to optimize the AHT agent's returns when dealing with a set of teammate policies seen during training.

Adversarial Diversity Unlike the aforementioned AHT methods, our work assumes no knowledge of the potentially encountered teammate policies. Instead, our goal is to learn what set of teammate policies, when used in AHT training, maximizes the AHT agent's robustness against previously unseen teammates. Previous methods achieve this goal by optimizing *Adversarial Diversity* (Cui et al. 2023a; Charakorn, Manoonpong, and Dilokthanakul 2023; Rahman et al. 2023). Optimizing adversarial diversity maximizes *self-play* returns, which are the expected returns when a generated policy π^{-i} collaborates with its intended partner policy π^{i}. At the same time, adversarial diversity metrics also minimize *cross-play* returns, the expected returns when π^{-i} collaborates with the intended partner of another policy π^{j}. Creating teammate policies by optimizing adversarial diversity can be detrimental to AHT training for two reasons. First, minimizing cross-play returns can lead towards a self-sabotaging teammate policy, π^{-i}, that minimizes the returns when collaborating with anyone not behaving like its intended partner, π^{-i}. Learning to collaborate with a self-sabotaging π^{-i} is difficult since learning to achieve high collaborative returns is only possible when the AHT agent fortuitously executes the same sequence of actions as π^{i} during exploration. Second, we show in Section 6.4 and Appendix B that optimizing adversarial diversity will never yield teammate policies that lead towards the most robust AHT agent in certain environments.

Other Diversity-based Methods Introducing diversity in training partners' policies is one way to generate robust response policies in multi-agent systems. A popular line of methods leverages population-based training and frequent checkpointing (Strouse et al. 2021; Vinyals et al. 2019; Cui et al. 2023b; Bakhtin et al. 2022). These methods rely on random seeds to find diverse policies, resulting in no guarantee that the generated policies are sufficiently diverse. Other studies optimize various types of diversity metrics directly into reinforcement learning objectives or as constraints. Xing et al. (2021) introduce a target-entropy regularization to Q-learning to generate information-theoretically different teammates. MAVEN (Mahajan et al. 2019) maximizes the mutual information between the trajectories and latent variables to learn diverse policies for exploration. Lupu et al. (2021) propose generating policies with different trajectory distributions. Trajectory diversity, however, is not necessarily meaningful for diversifying teammate policies (Rahman et al. 2023), so we do not consider these methods as baselines in our work.

3 Problem Formulation

The interaction between agents in an AHT environment can be modeled as a decentralized partially observable Markov decision process (Dec-POMDP). A Dec-POMDP is defined by an 8-tuple, $\langle N, S, \{\mathcal{A}^i\}_{i=1}^{|N|}, P, R, \{\Omega^i\}_{i=1}^{|N|}, O, \gamma\rangle$, with state space S, discount rate γ, and each agent $i \in N$ having an action space $\mathcal{A}^i$ and observation space Ω^i. Each interaction episode between the AHT agent and its teammates starts at an initial state s_0 sampled from an initial state distribution $p_0(s)$. Denoting $\Delta(X)$ as the set of all probability distributions over set X, at each timestep t agent i cannot perceive s_t and instead receives an observation $o^i_t \in \Omega^i$ sampled from the observation function, $O : S \mapsto \Delta(\Omega^1 \times \cdots \times \Omega^{|N|})$. Each agent $i \in N$ then decides its action at t, a^i_t, based on its policy, $\pi^i(H^i_t)$, that is conditioned on the observation-action history of agent i, $H^i_t = \{o^i_{\leq t}, a^i_{<t}\}$. The action selected by each agent is then jointly executed as a joint action, $\mathbf{a}_t$. After executing $\mathbf{a}_t$, the environment state changes following the transition function, $P : S \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^{|N|} \mapsto \Delta S$, and each agent receives a common scalar reward, r_t, according to the reward function, $R : S \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^{|N|} \mapsto \mathbb{R}$.

Existing AHT methods learn policies for a robust AHT agent by interacting with teammate policies from the training teammate policy set, $\Pi^{\text{train}} = \{\pi^{-1}, \pi^{-2}, \dots, \pi^{-K}\}$. The AHT agent then optimizes its policy to maximize its returns in interactions with policies from Π^{train}. The objective of these existing AHT methods can be formalized as:

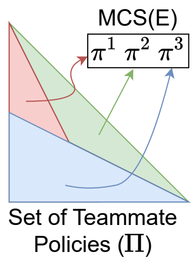
$$\pi^{*,i}(\Pi^{\text{train}}) = \underset{\pi^i}{\text{argmax}}\ \mathbb{E}_{\substack{\pi^{-i}\sim\mathbb{U}(\Pi^{\text{train}}),\\ a^i_t\sim\pi^i,\\ a^{-i}_t\sim\boldsymbol{\pi}^{-i},P,O}}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\right], \tag{1}$$

with $\mathbb{U}(X)$ denoting a uniform distribution over set X. The learned AHT agent policy, $\pi^{*,i}(\Pi^{\text{train}})$, is then evaluated for its robustness. Given an evaluated $\pi^{*,i}(\Pi^{\text{train}})$, this robustness measure, $M_{\Pi^{\text{eval}}}\left(\pi^{*,i}(\Pi^{\text{train}})\right)$, evaluates the expected returns when the AHT agent deals with teammates uniformly sampled from a previously unseen set of teammate policies, Π^{eval}. We formally define $M_{\Pi^{\text{eval}}}\left(\pi^{*,i}(\Pi^{\text{train}})\right)$ as the following expression:

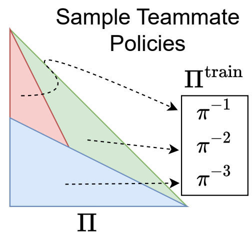
$$\mathbb{E}_{\substack{\boldsymbol{\pi}^{-i}\sim\mathbb{U}(\Pi^{\text{eval}}),a^i_t\sim\pi^{*,i}(\Pi^{\text{train}}),\\ a^{-i}_t\sim\boldsymbol{\pi}^{-i},P,O}}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\right], \tag{2}$$

The dependence of $\pi^{*,i}(\Pi^{\text{train}})$ on Π^{train} then implies that Expression 2 is also determined by Π^{train}.

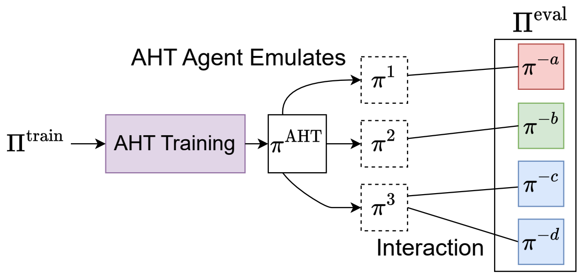
The goal of an AHT teammate generation process is to find Π^{train} producing an AHT agent policy that maximizes Expression 2 amid unknown Π^{eval}. Given the objective of AHT



(a) Best-response policies to each $\pi^{-i} \in \Pi$.



(b) Generating Π^{train} based on identified best-response policies.



(c) AHT training against Π^{train} and the expected results when dealing with previously unseen teammate policies.

Figure 1: Leveraging MCS(E) for Generating Robust AHT Agents. Figure 1a visualizes how teammate policies (points in the large triangle) can be grouped based on their best-response policies. The rectangle then shows an example MCS(E). From each subset of Π sharing the same best-response policy (colored small triangles), Figure 1b visualize how one policy is sampled from each subset to create Π^{train} for AHT training. As visualized in Figure 1c, using our generated Π^{train} for AHT training should encourage agents that emulate the best-response policy (dashed squares) to any $\pi^{-i} \in \Pi$ when dealing teammates from Π^{eval} (squares whose color represent its best-response policy).

training from Equation 1 and the definition of the robustness measure from Expression 2, the objective of an AHT teammate generation process is to find the optimal set of training teammate policies, $\Pi^{*,\text{train}}$, formalized as:

$$\underset{\Pi^{\text{train}}}{\text{argmax}}\, \mathbb{E}_{\Pi^{\text{eval}} \sim \mathbb{U}(\Pi)} \left[M_{\Pi^{\text{eval}}} \left(\pi^{*,i}(\Pi^{\text{train}}) \right) \right], \tag{3}$$

While uniformly sampling Π^{train} from Π may appear to be a reasonable solution to produce $\Pi^{*,\text{train}}$, training an AHT agent using Π^{train} may produce low returns if we only sample a limited number of policies from Π. When Π contains many possible teammate policies, the exact policies included in Π^{train} becomes important to ensure that the AHT agent is robust when collaborating with any teammate policy in Π.

4 Creating Robust AHT Agents By Identifying Minimum Coverage Sets

Assuming knowledge of Π^{eval}, the robustness of an AHT agent as defined by Expression 2 can be optimized by using Π^{eval} as teammate policies for AHT training. Given a teammate modeling component that accurately infers an unknown teammate's policy from Π^{eval} and an action selection component that can emulate any policy in the set of best-response policies to policies in Π^{eval}, $\text{BR}(\Pi^{\text{eval}})$, an AHT agent's robustness is maximized by following the best-response policy to the inferred teammate policy. Unfortunately, Π^{eval} being unknown makes this ideal training process impossible.

Improving an AHT agent's robustness without knowing Π^{eval} is still possible by identifying the *coverage set* of an environment. Denoting an environment characterized by a Dec-POMDP as E, any set containing at least one best-response policy to each teammate policy in Π is a coverage set of an environment, CS(E). CS(E) is formally characterized as:

$$\begin{aligned} &\forall \pi^{-i} \in \Pi, \forall H_t, \exists \pi^* \in \text{CS(E)} : \\ &\mathbb{E}_{s_0 \sim p_0} \left[\mathbf{R}_{*,-i}(H_t) \right] = \max_{\pi^i \in \Pi} \mathbb{E}_{s_0 \sim p_0} \left[\mathbf{R}_{,i,-i}(H_t) \right], \end{aligned} \tag{4}$$

where $\mathbf{R}_{i,-i}(H)$ denotes the following expression:

$$\mathbb{E}_{\substack{a^i_T \sim \pi^i(.|H_T), \\ a^{-i}_T \sim \pi^{-i}(.|H_T), \\ P,O}} \left[\sum_{T=t}^{\infty} \gamma^{T-t} R_T(s_T, a_T) \middle| H_t = H \right]. \tag{5}$$

Given this definition, a CS(E) remains a coverage set when policies are added. Thus, Π itself is trivially a coverage set.

Irrespective of Π^{eval}, CS(E) will contain at least a single best-response policy to any $\pi^{-i} \in \Pi^{\text{eval}}$ since $\Pi^{\text{eval}} \subseteq \Pi$. An AHT agent capable of emulating any policy from CS(E) consequently can follow any policy from $\text{BR}(\Pi^{\text{eval}})$ for any Π^{eval}. Therefore, training an AHT agent to emulate any policy from CS(E) gives us a solution to design robust AHT agents even when Π^{eval} is unknown.

Considering CS(E) may contain policies that are not a best-response policy to any member of Π, we ideally only train AHT agents to emulate a subset of CS(E) that consists of policies that are the best-response to some $\pi^{-i} \in \Pi$. Based on this idea, we define the *minimum coverage set* of an environment, MCS(E) $\subseteq \Pi$, that is a coverage set ceasing to be a coverage set if any of its elements are removed. This characteristic of MCS(E) is formalized as:

$$\forall \pi^i \in \text{MCS(E)} : \text{MCS(E)} - \{\pi^i\} \text{ is not a coverage set.} \tag{6}$$

In the example provided in Figure 1a, MCS(E) $= \{\pi^1, \pi^2, \pi^3\}$ is an MCS since the elimination of any policy, π, from it cause a subset of Π to not have their best-response policy in MCS(E) $- \{\pi\}$.

Our work aims to design AHT agents capable of emulating any policies from MCS(E) by constructing Π^{train} in a specific way. If Π^{train} is constructed for each $\pi^i \in$ MCS(E) to have a $\pi^{-i} \in \Pi^{\text{train}}$ such that $\pi^i \in \text{BR}(\{\pi^{-i}\})$, using Π^{train} while optimizing Equation 1 enables us to achieve this goal. The role of MCS(E) in our teammate generation process is visualized in Figures 1b and 1c.

5 L-BRDiv: Generating Teammate Policies By Approximating Minimum Coverage Sets

This section introduces our proposed teammate generation method based on estimating MCS(E). Section 5.1 details a constrained objective we use to estimate MCS(E). Finally, Section 5.2 provides a method that solves the constrained objective to jointly estimate MCS(E) while generating Π^{train}.

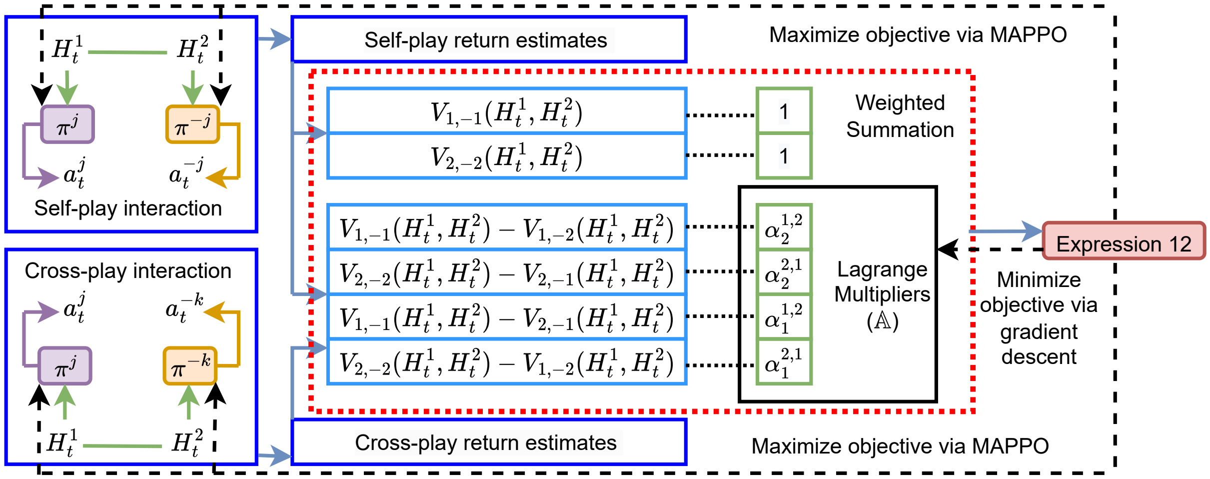


Figure 2: Lagrangian Best Response Diversity (L-BRDiv). The L-BRDiv algorithm trains a collection of policy networks (purple and orange boxes) and Lagrange multipliers (green cells inside the black rectangle). The purple boxes represent a policy from $\{\pi^i\}_{i=1}^K \subseteq \Pi$ while the policies visualized as an orange box is from $\{\pi^{-i}\}_{i=1}^K \subseteq \Pi$. Estimated returns between any possible pairs of policy, $(\pi^j, \pi^{-k}) \in (\{\pi^i|\pi^i \in \Pi\}_{i=1}^K \times \{\pi^{-i}|\pi^{-i} \in \Pi\}_{i=1}^K)$, and their associated Lagrange multipliers are used to compute the optimized term in the Lagrangian dual form (right red box) via a weighted summation operation (black dotted lines connect weights and multiplied terms). The policy networks are then trained via MAPPO (Yu et al. 2022) to maximize this optimized term, while the Lagrange multipliers are trained to minimize the term via stochastic gradient descent.

5.1 Jointly Approximating MCS(E) and Generating Π^{train}

Discovering MCS(E) by enumerating the AHT agent's best-response policy to each teammate policy is intractable given the infinite policies in Π. Instead, we can estimate MCS(E) by eliminating policies from a finite CS(E) to generate MCS(E). Given a finite CS(E), an AHT agent policy is not a member of MCS(E) if it is not the best response to any teammate policy.

We check if $\pi^i \in \text{CS(E)}$ is the best-response policy of at least one policy from Π by solving the *feasibility problem*, which is the following constrained optimization problem:

$$\max_{\pi^{-i} \in \Pi} \mathbb{E}_{s_0 \sim p_0} \left[\mathbf{R}_{i,-i}(H_t)\right], \tag{7}$$

with the following constraints:

$$\begin{aligned} &\forall \pi^j \in (\text{CS(E)} - \{\pi^i\}) : \\ &\mathbb{E}_{s_0 \sim p_0} \left[\mathbf{R}_{j,-i}(H_t)\right] \leq \mathbb{E}_{s_0 \sim p_0} \left[\mathbf{R}_{i,-i}(H_t)\right]. \end{aligned} \tag{8}$$

Any CS(E) member that violates the above constraint for all $\pi^{-i} \in \Pi$ is not a member of MCS(E). While this approach relies on knowing a finite CS(E), note that knowledge of a finite CS(E) is sometimes available. For instance, the set of all deterministic policies is a finite CS(E) for environments with a finite action space and state space.

Applying the above procedure to find MCS(E) can still be impossible for two reasons. First, a finite CS(E) can be unknown. Second, the size of CS(E) may be prohibitively large, which prevents solving the feasibility problem for all $\pi^i \in \text{CS(E)}$. Amid these challenging problems, we resort to estimating MCS(E) by only discovering its subset with K policies, $\text{MCS}^{\text{est}}(\text{E}) = \{\pi^i\}_{i=1}^K$.

We now describe an alternative constrained optimization objective that jointly finds $\text{MCS}^{\text{est}}(\text{E})$ while generating a set of teammate policies for AHT training, $\Pi^{\text{train}} = \{\pi^{-i}\}_{i=1}^K$, according to the method illustrated in Figure 1. Two characteristics are desired when finding $\text{MCS}^{\text{est}}(\text{E})$. First, we require each AHT agent policy from $\text{MCS}^{\text{est}}(\text{E})$ to only be the best-response policy to one teammate policy from Π^{train}, π^i. The second characteristic prioritizes the discovery of MCS(E) members that enables the AHT agent to produce high returns with a designated teammate policy, $\pi^{-i} \in \Pi$. These two requirements are formulated as the following constrained optimization problem:

$$\max_{\substack{\{\pi^i\}_{i=1}^K \subseteq \Pi, \\ \{\pi^{-i}\}_{i=1}^K \subseteq \Pi}} \sum_{i \in \{1,2,\dots,K\}} \mathbb{E}_{s \sim p_0} \left[\mathbf{R}_{i,-i}(H_t)\right], \tag{9}$$

with the following constraints that must be fulfilled for all $i, j \in \{1, 2, \dots, K\}$ and $i \neq j$:

$$\mathbb{E}_{s \sim p_0} \left[\mathbf{R}_{j,-i}(H_t)\right] + \tau \leq \mathbb{E}_{s \sim p_0} \left[\mathbf{R}_{i,-i}(H_t)\right], \tag{10}$$

$$\mathbb{E}_{s \sim p_0} \left[\mathbf{R}_{i,-j}(H_t)\right] + \tau \leq \mathbb{E}_{s \sim p_0} \left[\mathbf{R}_{i,-i}(H_t)\right]. \tag{11}$$

Note that a near-zero positive threshold ($\tau > 0$) is introduced in the constraints to prevent discovering duplicates of the same π^i and π^{-i}, which turns Constraints 10 & 11 into equality when $\tau = 0$.

5.2 Lagrangian BRDiv (L-BRDiv)

We present the **L**agrangian **B**est **R**esponse **Div**ersity (L-BRDiv) algorithm to generate Π^{train} that encourages an AHT agent to emulate $\text{MCS}^{\text{est}}(\text{E})$. L-BRDiv generates Π^{train} by solving the Lagrange dual of the optimization problem specified by Expressions 9-11, which is an unconstrained objective with the same optimal solution. The Lagrange dual for our optimization problem is defined as:

$$\min_{\substack{\mathbb{A}\subseteq\mathbb{R}_{\geq 0}^{K(K-1)}\\ \times\mathbb{R}_{\geq 0}^{K(K-1)}}} \max_{\substack{\{\pi^i\}_{i=1}^K\subseteq\Pi,\\ \{\pi^{-i}\}_{i=1}^K\subseteq\Pi}} \Bigg(\sum_{i\in\{1,\dots,K\}} \mathbb{E}_{s_0\sim p_0}\left[\mathbf{R}_{i,-i}(H_t)\right] + \sum_{\substack{i,j\in\{1,\dots,K\}\\ i\neq j}} \alpha_1^{i,j}\left(\mathbb{E}_{s_0\sim p_0}\left[\mathbf{R}_{i,-i}(H_t)-\tau-\mathbf{R}_{j,-i}(H_t)\right]\right) + \sum_{\substack{i,j\in\{1,\dots,K\}\\ i\neq j}} \alpha_2^{i,j}\left(\mathbb{E}_{s_0\sim p_0}\left[\mathbf{R}_{i,-i}(H_t)-\tau-\mathbf{R}_{i,-j}(H_t)\right]\right)\Bigg), \tag{12}$$

with $\mathbb{A}=\{(\alpha_1^{i,j},\alpha_2^{i,j})|\alpha_1^{i,j}\geq 0,\alpha_2^{i,j}\geq 0\}_{i,j\in\{1,2,\dots,K\},i\neq j}$ denoting the set of optimizable Lagrange multipliers.

L-BRDiv learns to assign different values to Lagrange multipliers in $\mathbb{A}$ of (12). Optimizing Lagrange multipliers gives L-BRDiv two advantages over previous methods, which treat these hyperparameters as constants. First, we demonstrate in Section 6 that L-BRDiv creates better Π^{train} by identifying more members of MCS(E). Second, it does not require hyperparameter tuning on appropriate weights associated with cross-play return, which in previous methods require careful tuning to discover members of MCS(E) (Rahman et al. 2023) and prevent the generation of incompetent policies not achieving high returns against any AHT agent policy (Charakorn, Manoonpong, and Dilokthanakul 2023).

We detail L-BRDiv's teammate generation process in Algorithm 1 and analyze its computational complexity in Appendix D. L-BRDiv implements the policies optimized in the Lagrange dual as neural networks trained with MAPPO (Yu et al. 2022) to maximize the weighted advantage function (14), whose weights correspond to the total weight associated with each expected return term in (12). At the same time, L-BRDiv trains a critic network to bootstrap the evaluation of (12) instead of a Monte Carlo approach, which can be expensive since it requires all generated policy pairs to initially follow the observation-action history, H_t. Meanwhile, the Lagrange multipliers are trained to minimize (12) while ensuring it is non-negative. Figure 2 then summarizes the training process of L-BRDiv's models.

Algorithm 1: Lagrangian Best Response Diversity

Cardinality of MCS^{est}(E) and Π^{train}, K.
Randomly initialized policy networks in MCS^{est}(E) & Π^{train}, denoted by $\{\pi^i_{\theta_i}\}_{i=1}^K$ & $\{\pi^{-i}_{\theta_{-i}}\}_{i=1}^K$ respectively.
Randomly initialized critic network $V^{j,-i}_{\theta_c}$, target $V^{j,-i}_{\theta'_c}$.
Initial values for the Lagrange multipliers, $\mathbb{A}$.
1: **for** $t_{\text{update}}=1,2,\dots,N_{\text{updates}}$ **do**
2: $(i,j)\sim\mathbb{U}(\{1,2,\dots,K\}^2)$
3: $D\leftarrow$ AgentInteraction$(\pi^j_{\theta_j},\pi^{-i}_{\theta_{-i}})$
4: **for** $(H_t,a_t,r_t,H_{t+1})\in D$ **do**
5: // Critic & Policy Optimization Step (Lines 6 & 8)
6: Update θ_c with SGD & a target critic to minimize

$$\left(V^{j,-i}_{\theta_c}(H_t)-r_t-\gamma V^{j,-i}_{\theta'_c}(H_{t+1})\right)^2 \tag{13}$$

7: $$w^{i,j}(\mathbb{A})\leftarrow\begin{cases}1+\sum\limits_{k\neq j}\left(\alpha_1^{i,k}+\alpha_2^{i,k}\right) & ,i=j\\ -\left(\alpha_1^{i,j}+\alpha_2^{j,i}\right) & ,i\neq j\end{cases}$$

8: Update θ_j and θ_{-j} with MAPPO to maximize:

$$w^{i,j}(\mathbb{A})\left(r_t+\gamma V^{j,-i}_{\theta_c}(H_{t+1})-V^{j,-i}_{\theta_c}(H_t)\right) \tag{14}$$

9: **if** t_{update} mod T_{lagrange} = 0 **then**
10: // Lagrange Multiplier Optimization Step
11: Update $\mathbb{A}$ using SGD to minimize Expression 12 where $\forall i,j\in\{1,2,\dots,K\}$:

$$\mathbb{E}_{s_0\sim p_0}\left[\mathbf{R}_{j,-i}(H_t)\right]\approx V^{j,-i}_{\theta_c}(H_t) \tag{15}$$

12: $\mathbb{A}\leftarrow\{\max(\alpha,0)\mid\alpha\in\mathbb{A}\}$
13: **end if**
14: **end for**
15: **end for**
16: **Return** $\{\pi^{-i}_{\theta_{-i}}\}_{i=1}^K$

6 Experiments

In this section, we describe the environments and baseline algorithms in Sections 6.1 and 6.2. Section 6.3 then details the experiment setups for evaluating the robustness of AHT agents in L-BRDiv and baseline methods via their generated training teammate policies. Finally, we present the AHT experiment results and an analysis of MCS^{est}(E) policies identified by L-BRDiv in Sections 6.4 and 6.5.

6.1 Environments

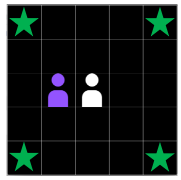
We run our experiments in four two-player cooperative environments. The first environment is a repeated matrix game where agents have three actions, whose reward function is provided in Figure 3a. Since eliminating self-sabotaging behaviour (Cui et al. 2023a) is not the focus of our work, we remove teammate-related information and actions from an agent's observation such that self-sabotaging behaviour is not a member of possibly discovered teammate behaviours, Π. We also do experiments in the Cooperative Reaching environment (Rahman et al. 2023) where two agents can move across the four cardinal directions in a two-dimensional grid world. Both agents are given a reward of 1 once they simultaneously arrive at the same corner grid. The third environment is Weighted Cooperative Reaching, which is similar to Cooperative Reaching except for a modified reward function (Figure 3c) that provides lower rewards if both agents arrive at different corner cells. The last environment is Level-based Foraging (LBF) (Christianos, Schäfer, and Albrecht 2020), where both agents must move along the four cardinal directions to a cell next to the same object and retrieve it by simultaneously selecting actions for collecting objects. Successful object collection gives both agents a reward of 0.33.

6.2 Baseline Methods

Our experiments compare L-BRDiv against methods that maximize adversarial diversity, such as BRDiv (Rahman

10	0	4
0	6	4
4	4	6

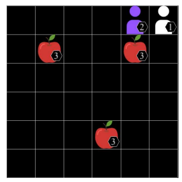
(a) Repeated Matrix Game.



(b) Coop Reaching.

	A	B	C	D
A	10	0	6	6
B	0	10	6	6
C	6	6	8	0
D	6	6	0	8

(c) Weighted Coop Reaching.



(d) LBF.

Figure 3: Environments Used in AHT Experiments. We provide experiments in a repeated matrix game whose reward function is displayed in Figure 3a. Figure 3b displays an example state of the Cooperative Reaching environment where the green stars represent corner cells that provide agents rewards once they simultaneously reach it. If we start from the top-left corner cell in Figure 3b and assign IDs (A-D) to corner cells in a clockwise manner, Figure 3c shows the reward function of the Weighted Cooperative Reaching environment where agents' rewards depend on which pair of destination cells the two agents arrive at. Finally, Figure 3d shows a sample state of Level-based Foraging (LBF) where the apples represent the collected objects.

et al. 2023) and LIPO (Charakorn, Manoonpong, and Dilokthanakul 2023). Comparing L-BRDiv and BRDiv helps investigate the detrimental effect of using fixed uniform weights instead of L-BRDiv's optimized Lagrange multipliers ($\mathbb{A}$). Meanwhile, including LIPO as a baseline enables us to investigate the advantage of L-BRDiv and BRDiv's use of weights with a larger magnitude for self-play maximization (i.e. $w^{i,i}(\mathbb{A})$ in Eq. 14) compared to the weights for cross-play minimization (i.e. $w^{i,j}(\mathbb{A})$ in Eq. 14). We do not compare our method with ADVERSITY (Cui et al. 2023a), which combines LIPO with techniques to prevent self-sabotage. We hold that self-sabotaging policies should not be ruled out during policy generation since teammates may still use them. By not preventing the discovery of such policies, we ensure that our method remains fully general.

6.3 Experiment Setup

We start our experiments for each environment by generating K training teammate policies using the compared methods. We ensure fairness in our experiments by using RL^2 algorithm (Duan et al. 2016) to find an optimal AHT agent policy defined in Equation 1 based on Π^{train} generated by each teammate generation algorithm. Since our partially observable environments provide no useful information to infer teammate policies except for rewards obtained at the end of each interaction episode, we choose RL^2 since it can use reward information to create agent representations maintained and updated across multiple episodes. For each of the compared algorithms, the teammate generation and AHT training process are repeated under four seeds to allow for a statistically sound comparison between each method's performance. As a measure of robustness, we then evaluate the average returns of the AHT agent trained from each experiment seed when collaborating with policies sampled from Π^{eval}. We construct Π^{eval} for each environment by creating heuristic-based agents, whose behaviour we describe in Appendix A. Finally, we compute the mean and 95% confidence interval of the recorded returns across four seeds and report it in Figure 4.

6.4 Ad Hoc Teamwork Experiment Results

Figure 4 shows the results of the AHT experiments. We find that L-BRDiv significantly outperforms other compared methods in the repeated matrix game, Weighted Cooperative Reaching, and LBF. While BRDiv slightly outperforms L-BRDiv in Cooperative Reaching, overlapping confidence intervals among the last few checkpoints suggest that the difference is only marginally significant.

L-BRDiv outperforms the compared baselines in all environments except Cooperative Reaching since these environments all have reward functions that cause some members of the MCS, $\pi^i \in$ MCS(E), to yield high expected returns in cross-play interactions against a generated teammate policy, $\pi^{-j} \in \Pi^{\text{train}}$, that is not its intended partner, $\pi^{-i} \in \Pi^{\text{train}}$. Meanwhile, all $\pi^i \in$MCS(E) for Cooperative Reaching have equally low (i.e. zero) returns against the intended partner of other MCS(E) members. The large cross-play returns disincentivize BRDiv and LIPO's optimized objective from discovering π^i and π^{-i} during teammate generation. The inability to discover $\pi^i \in$ MCS(E) and π^{-i} will then lead towards diminished robustness since the trained AHT agent will yield lower returns against teammates whose best-response policy is π^i. In contrast, Cooperative Reaching's reward structure makes MCS(E) (i.e. the set of four policies moving towards each distinct corner cell) consist of policies yielding equally low cross-play returns of zero among each other.

Although both BRDiv and LIPO are equipped with a hyperparameter, $\alpha > 0$, that can change weights associated with self-play returns maximization and cross-play returns minimization in their learning objective, it is possible to find simple scenarios where no feasible α facilitates the discovery of a desirable Π^{train} to maximize an AHT agent's robustness. Such a desirable Π^{train} is characterized by all AHT agent policies in MCS(E) having at least one teammate policy in $\in \Pi^{\text{train}}$ whom it is the best-response policy to. Appendix B shows that the Repeated Matrix Game and Weighted Cooperative Reaching environment are examples of such scenarios. Even in environments like LBF where there may exist an α enabling both BRDiv and LIPO to discover a desirable Π^{train} by optimizing their learning objectives, finding an appropriate α is costly if we factor in the computational resources required to run a single teammate generation process. Unlike BRDiv and LIPO, L-BRDiv's inclusion of Lagrange multipliers as learned parameters enables it to discover desirable Π^{train} in a wider range of environments while reducing the number of hyperparameters that must be tuned.

Note that L-BRDiv and the baseline methods all successfully discover MCS(E) in Cooperative Reaching. However,

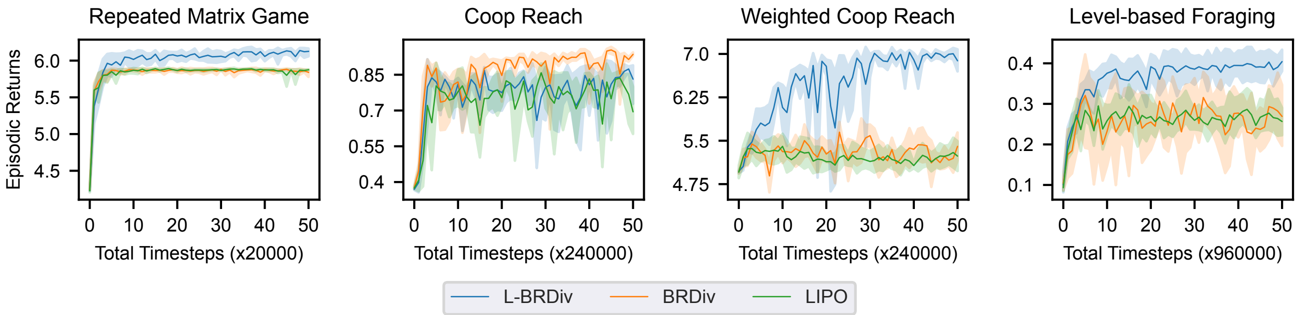
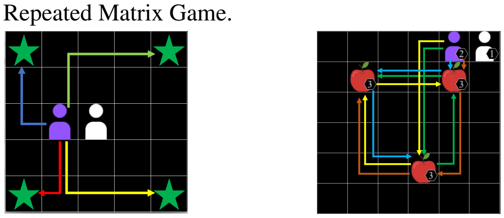


Figure 4: Generalization Performance Against Previously Unseen Teammate Types. This figure shows that L-BRDiv produced significantly higher episodic returns when dealing with unknown teammate policies in all environments except for Cooperative Reaching. We also show L-BRDiv achieving similar returns to other methods in Cooperative Reaching.

	π(A)	π(B)	π(C)
1	1	0	0
2	0	1	0
3	0	0	1

(a) AHT agent action selection probability for policies in MCS^{est}(E) in the Repeated Matrix Game.



(b) MCS^{est}(E) in Coop Reaching & Weighted Coop Reaching.

(c) AHT agent policies in the MCS^{est}(E) discovered for LBF.

Figure 5: MCS^{est}(E) Yielded by L-BRDiv. L-BRDiv is capable of estimating all members of MCS(E) in all environments except LBF. Even so, L-BRDiv still discovers more conventions with distinct best-response policies than the baselines in LBF. The discovery of more MCS(E) results in L-BRDiv producing more robust AHT agents.

each teammate policy generated by L-BRDiv and LIPO which has one of the MCS(E) members as its best-response policy ends up being less optimal than their BRDiv-generated counterparts. These suboptimal policies require more steps to complete an episode by occasionally moving away from their destination corner cell. Learning from these suboptimal agents made the AHT agent less decisive when selecting which corner cell to move towards and finally ends up producing agents with slightly lower returns.

6.5 Behaviour Analysis

The AHT agent policies that L-BRDiv discovers as members of MCS^{est} in all environments are provided in Figures 5a-5c. Unlike the compared baseline methods that only discover two members of MCS(E), results from the Repeated Matrix Game show L-BRDiv is capable of consistently finding all three deterministic policies that are members of MCS(E). As a consequence of Cooperative Reaching's reward structure, all compared methods successfully discover MCS(E) and achieve the similar performances. Meanwhile, L-BRDiv is the only method that finds all four members of MCS(E) corresponding to movement towards each corner grid in Weighted Cooperative Reaching. As we show in Appendix B, BRDiv and LIPO's failure to discover all members of MCS(E) in the Repeated Matrix Game and Weighted Cooperative Reaching is because discovering MCS(E) does not optimize their optimized objective for any constant and uniform α. Despite no method perfectly discovering MCS(E) consisting of all six possible orderings for collecting objects in LBF, L-BRDiv is closer to estimating MCS(E) than the baseline algorithms by discovering four MCS(E) members in one seed and five MCS(E) members in the remaining seeds. L-BRDiv's ability to discover more MCS(E) members than baselines leads towards more robust AHT agents that can emulate the best-response policy to a wider range of teammate policies.

7 Conclusion & Future Work

In this work, we propose that an appropriate set of teammate policies for AHT training must enable agents to emulate all policies in MCS(E), the smallest set of policies containing the best-response policy to any teammate policy in Π. To generate such teammate policies for robust AHT training, we introduce and evaluate L-BRDiv. By solving a constrained optimization problem using the Lagrange multiplier technique, L-BRDiv then learns to jointly approximate the MCS of an environment and generate a set of teammate policies for AHT training. Our experiments indicate that L-BRDiv yields more robust AHT agents compared to state-of-the-art teammate generation methods by identifying more members of the MCS while also removing the need for tuning important hyperparameters used in prior methods.

Future work will consider extending L-BRDiv to more complex environments where more than two agents must collaborate. Another promising research direction is to extend L-BRDiv with techniques to discourage the discovery of self-sabotaging policies (Cui et al. 2023a). Finally, applying our method in fully competitive and general-sum games is another promising direction for creating robust agents since the concept of minimum coverage sets is not limited to fully cooperative problems.

Acknowledgements

All research conducted in this work was done under the Learning Agents Research Group (LARG) at UT Austin's Department of Computer Science. The research in LARG is supported in part by Lockheed Martin, NSF (CPS-1739964, IIS-1724157, NRI-1925082), ONR (N00014-18-2243), FLI (RFP2-000), ARO (W911NF19-2-0333), DARPA, GM, and Bosch. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin following its policy on objectivity in research.

References

Bakhtin, A.; Wu, D. J.; Lerer, A.; Gray, J.; Jacob, A. P.; Farina, G.; Miller, A. H.; and Brown, N. 2022. Mastering the Game of No-Press Diplomacy via Human-Regularized Reinforcement Learning and Planning. *arXiv preprint arXiv:2210.05492*.

Barrett, S.; Rosenfeld, A.; Kraus, S.; and Stone, P. 2016. Making Friends on the Fly: Cooperating with New Teammates. *Artificial Intelligence*.

Chakraborty, D.; and Stone, P. 2014. Convergence, targeted optimality and safety in multiagent learning. *Sample Efficient Multiagent Learning in the Presence of Markovian Agents*, 29–47.

Charakorn, R.; Manoonpong, P.; and Dilokthanakul, N. 2023. Generating Diverse Cooperative Agents by Learning Incompatible Policies. In *The Eleventh International Conference on Learning Representations*.

Christianos, F.; Schäfer, L.; and Albrecht, S. V. 2020. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Cui, B.; Lupu, A.; Sokota, S.; Hu, H.; Wu, D. J.; and Foerster, J. N. 2023a. Adversarial Diversity in Hanabi. In *The Eleventh International Conference on Learning Representations*.

Cui, J.; Yang, X.; Luo, M.; Lee, G.; Stone, P.; Lee, H.-H. S.; Lee, B.; Suh, G. E.; Xiong, W.; and Tian, Y. 2023b. MACTA: A Multi-agent Reinforcement Learning Approach for Cache Timing Attacks and Detection. In *The Eleventh International Conference on Learning Representations*.

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. RL^2: Fast reinforcement learning via slow reinforcement learning.

Gu, P.; Zhao, M.; Hao, J.; and An, B. 2021. Online ad hoc teamwork under partial observability. In *International Conference on Learning Representations*.

Hu, H.; Lerer, A.; Peysakhovich, A.; and Foerster, J. 2020. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, 4399–4410. PMLR.

Lupu, A.; Cui, B.; Hu, H.; and Foerster, J. 2021. Trajectory diversity for zero-shot coordination. In *International conference on machine learning*, 7204–7213. PMLR.

Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32.

Mirsky, R.; Carlucho, I.; Rahman, A.; Fosong, E.; Macke, W.; Sridharan, M.; Stone, P.; and Albrecht, S. V. 2022. A survey of ad hoc teamwork research. In *European Conference on Multi-Agent Systems*, 275–293. Springer.

Papoudakis, G.; Christianos, F.; and Albrecht, S. 2021. Agent Modelling under Partial Observability for Deep Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35.

Rahman, A.; Fosong, E.; Carlucho, I.; and Albrecht, S. V. 2023. Generating Teammates for Training Robust Ad Hoc Teamwork Agents via Best-Response Diversity. *Transactions on Machine Learning Research*.

Rahman, A.; Höpner, N.; Christianos, F.; and Albrecht, S. V. 2021. Towards Open Ad Hoc Teamwork Using Graph-Based Policy Learning. In *International Conference on Machine Learning*, volume 139. PMLR.

Stone, P.; Kaminka, G.; Kraus, S.; and Rosenschein, J. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 1504–1509.

Strouse, D.; McKee, K.; Botvinick, M.; Hughes, E.; and Everett, R. 2021. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34: 14502–14515.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Xing, D.; Liu, Q.; Zheng, Q.; Pan, G.; and Zhou, Z. 2021. Learning with Generated Teammates to Achieve Type-Free Ad-Hoc Teamwork. In *IJCAI*, 472–478.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zintgraf, L.; Devlin, S.; Ciosek, K.; Whiteson, S.; and Hofmann, K. 2021. Deep interactive bayesian reinforcement learning via meta-learning. *arXiv preprint arXiv:2101.03864*.