Quantifying Interaction Level Between Agents Helps Cost-efficient Generalization in Multi-agent Reinforcement Learning

Yuxin Chen
yuxinc@berkeley.edu
University of California, Berkeley

Chen Tang
chen.tang@austin.utexas.edu
The University of Texas at Austin

Ran Tian
rantian@berkeley.edu
University of California, Berkeley

Chenran Li
chenran_li@berkeley.edu
University of California, Berkeley

Jinning Li
jinning_li@berkeley.edu
University of California, Berkeley

Masayoshi Tomizuka
tomizuka@berkeley.edu
University of California, Berkeley

Wei Zhan
wzhan@berkeley.edu
University of California, Berkeley

Abstract

Generalization poses a significant challenge in Multi-agent Reinforcement Learning (MARL). The extent to which unseen co-players influence an agent depends on the agent's policy and the specific scenario. A quantitative examination of this relationship sheds light on how to effectively train agents for diverse scenarios. In this study, we present the *Level of Influence* (LoI), a metric quantifying the interaction intensity among agents within a given scenario and environment. We observe that, generally, a more diverse set of co-play agents during training enhances the generalization performance of the ego agent; however, this improvement varies across distinct scenarios and environments. LoI proves effective in predicting these improvement disparities within specific scenarios. Furthermore, we introduce a LoI-guided resource allocation method tailored to train a set of policies for diverse scenarios under a constrained budget. Our results demonstrate that strategic resource allocation based on LoI can achieve higher performance than uniform allocation under the same computation budget. The code is available at: https://github.com/ThomasChen98/Level-of-Influence.

1 Introduction

Creating agents capable of effectively interacting with other agents, in particular humans, has been a longstanding challenge (Bard et al., 2020; Dafoe et al., 2020). Agents trained with model-free reinforcement learning (RL) have shown the potential to reach or surpass human-level performance through self-play (SP) in both classical discrete board games (Silver et al., 2017; 2018; Zha et al., 2021) and continuous domains such as Dota (Berner et al., 2019), Starcraft (Vinyals et al., 2019), and racing (Fuchs et al., 2021). However, SP agents typically undergo training with replicas of themselves, resulting in limited adaptability and robustness when interacting with previously unseen

co-players exhibiting different behaviors (Lowe et al., 2020; Bullard et al., 2020; Strouse et al., 2021; McKee et al., 2022).

One promising solution to improve the policy robustness is *diversifying the co-player distribution*. It has been shown that computationally hard problems like chess and go could benefit from diversifying agents during training (Zahavy et al., 2023). Prior studies introduced and validated methods for more complex games, such as population-based training (Jaderberg et al., 2019; Carroll et al., 2019; Jaderberg et al., 2017), league-based training (Vinyals et al., 2019), fictitious self-play (Heinrich et al., 2015; Strouse et al., 2021), and diversification of agent hyperparameters (Hu et al., 2020; McKee et al., 2020). However, it is important to note that a trade-off exists in most of these methods, where enhancing generalization capabilities comes at the cost of increased training resources and time.

Nevertheless, an important question remains: *is diversifying the co-player distribution during training always worthwhile?* In practice, various real-world applications require a set of tailored RL policies for diverse target scenarios (Lowe et al., 2017; Fuchs et al., 2021). Diversifying the co-player distribution during training in all the target scenarios comes at a high training cost, with the resulting benefits varying across scenarios. In particular, we argue that the benefits of introducing diverse co-players depend on *how intensive the agents interact in the specific scenario*. For instance, consider training autonomous driving agents. Enhanced generalization does not provide substantial advantages on highways as it does in crowded intersections and roundabouts. On highways, vehicles focus on lane keeping most of the time, involving fewer interactions. In contrast, in roundabouts and intersections where agents' trajectories are highly interdependent, the presence of diverse surrounding agent behaviors has a much more significant impact on the ego agent policy.

Our key insight is that, by quantifying the interaction intensity, we can assess the necessity of diversifying co-player policy distribution when training the ego agent policy as the effects of environmental variation, and allocate the training resources strategically to maximize the overall advantage.

To this end, we introduce the *Level of Influence* (LoI), a metric quantifying the interaction intensity among agents within a given scenario. We propose to quantify the interaction intensity by how much the ego agent's reward is affected by the variation of non-ego agents' behavior. Formally, inspired by (Jaques et al., 2019), we define LoI as the conditional mutual information (MI) between the ego agent's expected reward and the non-ego agent's policy selection. We validate the effectiveness of using LoI for cost-efficient generalization by training a set of policies with co-players of different levels of diversity for groups of scenarios and environments. We find that the LoI metric is highly correlated with the benefits of having diverse co-player policy distribution on the generalization of the ego agent within given scenarios, *i.e.*, a higher LoI value indicates that a larger improvement can be anticipated in the ego agent's performance when a more diverse set of co-play agents are encountered during training. Consequently, we design a LoI-guided resource allocation method to train a set of policies for diverse scenarios under a limited training budget. We compare the overall performance between the LoI-guided and uniform allocation schemes, showcasing that the LoI-guided scheme consistently yields higher average performance across a range of game settings. We summarize the novel contributions of this paper as follows:

1. We propose a novel metric, Level of Influence (LoI), to quantify the interaction among agents in general multi-agent reinforcement learning problems.
2. We demonstrate that the LoI metric is highly correlated with the benefits of having diverse co-player distribution on the generalization of the ego agent within given scenarios.
3. We propose a LoI-guided resource allocation method and show that it can achieve a higher average reward than uniform allocation under the same computation budget.

2 Related Work

Ad-hoc Teamwork. Ad-hoc teamwork (AHT) (Stone et al., 2010), also referred to as zero-shot coordination (ZSC) (Hu et al., 2020), involves training agents to collaborate with co-players they

have not encountered before. Early approaches primarily focused on game-theoretic analysis within matrix games (Stone et al., 2009; Agmon & Stone, 2012). Recently, multi-agent reinforcement learning (MARL) has enabled ad-hoc teamwork in more intricate grid worlds and continuous domains. Various works have explored hierarchical social intention (Kleiman-Weiner et al., 2016), social conventions (Shih et al., 2021), shared planning (Ho et al., 2016), and theory of mind (Choudhury et al., 2019) in this context. In MARL, an agent's learning is influenced by both other co-players and the environment (Littman, 1994). However, most of the previously mentioned works do not explicitly evaluate the impacts of environmental variations. Carroll et al. (Carroll et al., 2019) introduce the game Overcooked and explicitly showcase that the environment configurations affect the robustness of the trained agents when teamed with unknown human players. Subsequent research includes diverse layout generation (Fontaine et al., 2021; McKee et al., 2022) and scalable evaluation (Leibo et al., 2021). Nonetheless, these works only provide *qualitative* analyses of different environments and do not *quantitatively* measure such effects across scenarios.

Generalization in Multi-agent Reinforcement Learning. In the field of MARL, various attempts have been made to enhance agents' adaptability to new co-players. Jaderberg et al. (Jaderberg et al., 2017; 2019) introduced population-based training (PBT) to jointly optimize the performance of a population of models. Several variations of PBT include the league-based training that masters the full game of StarCraft II (Vinyals et al., 2019), the fictitious co-play (FCP) that can reach human-level performance (Heinrich et al., 2015; Strouse et al., 2021), and heterogeneous populations training with Social Value Orientation (SVO) (McKee et al., 2020). However, high-performing agents come at the cost of more expensive training cost. Considering the varying benefits of generalization across diverse environments, we aim to evaluate if the extra training cost for enhanced generalization is justified. This area is relatively under-explored in existing research.

Causal Influence. Our work shows a notable connection to Jaques et al. (Jaques et al., 2019), where a causal influence reward is incorporated as an intrinsic motivation during the training of RL agents. This reward incentivizes agents to maximize mutual information (MI) between their actions. The goal of maximizing MI between actions is to encourage more coordinated behavior among the agents. The causal influence is assessed using counterfactual reasoning (McAllister et al., 2022; Foerster et al., 2018; Pearl, 2013) where an agent simulates alternate, counterfactual *actions* that it could have taken at every time step. In contrast, we measure the mutual information between the ego agent's expected reward and the non-ego agent's policy selection by simulating counterfactual *policies* that the non-ego agent would have chosen within given scenarios.

3 Preliminaries

3.1 Multi-agent Markov Decision Process

An *n-player partially observable Markov game* $\mathcal{M}$ (Boutilier, 1996; McKee et al., 2022) is defined by tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{O}, \{\mathcal{A}_i\}_{i\in\alpha}, \mathcal{T}, \{r_i\}_{i\in\alpha} \rangle$, where $\mathcal{S}$ is the finite set of *states*, $\mathcal{O}\colon \mathcal{S} \times \{1, \ldots, n\} \mapsto \mathbb{R}^d$ is the *observation* function specifying each agent's d-dimensional view on the state space followed by their joint observation $\vec{o} = (o_1, \ldots, o_n)$. Let α be a finite set of *agents*, $\mathcal{A}_i$ is a finite set of discrete *actions* available to agent i. The joint action is defined as $\vec{a} = (a_1, \ldots, a_n) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$. The stochastic *transition function* $\mathcal{T}\colon \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_n \mapsto \Delta(\mathcal{S})$ determines the discrete probability distribution over the next state given the current state and the joint action. Each agent i receives its real-valued *reward* defined as $r_i\colon \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_n \mapsto \mathbb{R}$.

We assume that each agent learns their policy in a *decentralized* manner (*i.e.*, independently learns a *policy* $\pi_i(a_i|o_i)$ based on its own observation o_i by optimizing its own individual reward r_i) without direct communication. We use $\vec{\pi} = (\pi_1, \ldots, \pi_n)$ to denote the joint policy. Agent i optimizes for a policy that maximizes the long-term γ-discounted payoff (McKee et al., 2022) defined as

$$V_{\pi_i}(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_i(s_t, \vec{a}_t) | \vec{a}_t \sim \vec{\pi}_t, s_{t+1} \sim \mathcal{T}(s_t, \vec{a}_t)\right], \tag{1}$$

where γ denotes the discount factor discounting future rewards.

Algorithm 1 Level of Influence calculation

Input: # Alice policies a, # Bob policies b, # Alice checkpoints per policy m, # Bob checkpoints per policy n, Alice sampling probability P_φ, Bob sampling probability P_ϑ, # game per Alice-Bob pair g

Train a Alice policies with SP and save checkpoints pool Φ_i
Train b Bob policies with SP and save checkpoints pool Θ_j
Initialize set of index $\mathcal{I} \leftarrow \{\}$
for i=1:a **do**
  Sample m Alice checkpoints $\phi_{i,k} \sim \Phi_i$ with P_φ
  for k=1:m **do**
    for j=1:b **do**
      Sample n Bob checkpoints $\theta_{j,l} \sim \Theta_j$ with P_ϑ
      Initialize set of distribution $\mathcal{P} \leftarrow \{\}$
      for l=1:n **do**
        $P_{R_{i,k}|\vartheta=\theta_{j,l},\ \varphi=\phi_{i,k}} \leftarrow$ DISTRIBUTION($\phi_{i,k}$, $\theta_{j,l}$, g)
        $\mathcal{P} \leftarrow \mathcal{P} \cup P_{R_{i,k}|\vartheta=\theta_{j,l},\ \varphi=\phi_{i,k}}$
      end for
      $P_{R_{i,k}|\varphi=\phi_{i,k}} \leftarrow$MARGINAL_DISTRIBUTION($\mathcal{P}$, P_ϑ) ▷ Equation 3
      $I_{i,j,k} \leftarrow$MUTUAL_INFORMATION($P_{R_{i,k}|\varphi=\phi_{i,k}}$, $\mathcal{P}$) ▷ Equation 4d
      $\mathcal{I} \leftarrow \mathcal{I} \cup I_{i,j,k}$
    end for
  end for
end for
$\bar{I} \leftarrow$ AVERAGE($\mathcal{I}$)

Output: $\bar{I}$

function DISTRIBUTION(ϕ, θ, g)
  Initialize set of reward $\mathcal{R} \leftarrow \{\}$
  for i=1:g **do**
    Game between ϕ and θ to collect reward r
    $\mathcal{R} \leftarrow \mathcal{R} \cup r$
  end for
  $P_{R|\vartheta=\theta,\varphi=\phi} \leftarrow$ HISTOGRAM($\mathcal{R}$)
  return $P_{R|\vartheta=\theta,\varphi=\phi}$
end function

3.2 Multi-agent Reinforcement Learning

Self-play. Self-play (SP) is an online evolutionary algorithm in which agents learn by playing against duplicates of themselves. Policies trained via SP have succeeded in a variety of environments and game settings (Silver et al., 2018; Vinyals et al., 2019; Berner et al., 2019; Zha et al., 2021). In the SP training, all the agents are initialized with random policies, and we keep updating the ego agent's policy while fixing other agents' policies. Throughout the training phase, the policies of the ego agent are stored as checkpoints periodically. Subsequently, following each checkpoint save, all non-ego agents adopt the recently saved checkpoint as their updated policies. (*i.e.*, all non-ego agents become the latest duplicates of the ego agent). One major drawback of SP is that the learning agent can not generalize well to new partners deviating from its own policy distribution (Strouse et al., 2021; Bullard et al., 2020; 2021; Lowe et al., 2020), as agents only learn how to collaborate with themselves during training.

Population-play. Population-play (PP), on the other hand, keeps a population of agents training in parallel (Jaderberg et al., 2017). The environment and its agents are initialized with p different random seeds. A population of p policies is then trained from the p distinct initialization through

interacting with each other. Specifically, instead of loading the recently saved checkpoints from the ego agent's own training history, the non-ego agents are randomly selected from the remaining $p-1$ trained policies' latest checkpoints. By mutating the non-ego agents across the population, the trained PP agents acquire better generalization capabilities than SP (Jaderberg et al., 2017; Carroll et al., 2019; Strouse et al., 2021; McKee et al., 2022).

Training Algorithm. In MARL, agents' policies are parameterized by neural network models and can be trained with various deep RL algorithms. In our case, we use Proximal Policy Optimization (PPO) (Schulman et al., 2017) for model training.

4 Methodology

We aim to study the potential impact of the diversity of co-play agents during training on the generalization performance as the effects of environmental variation. In MARL, the performance of a policy is measured by its expected reward when pairing with different co-players, which is determined by the reward design (*i.e.*, payoff matrix). We refer to games that have distinct reward designs as *environments*. Under the same reward design, one can create different variations of the game by changing the map layout (*e.g.*, size, shape, obstacle locations, etc.). We refer to the games with distinct map layouts within the same *environment* as *scenarios*. We hypothesize that enhanced generalization yields varied levels of performance improvements for the agent in different scenarios. Intuitively, this is because agents in different scenarios have different interaction intensities, and the generalization performance, as measured by the expected reward, depends greatly on the interaction pattern and frequency. Therefore, we aim to find a metric that quantifies the interaction between agents and examine it as an indicator of the potential generalization improvement by having a more diverse set of co-player policies during training.

4.1 Level of Influence

In order to quantitatively describe the interaction intensity between each agent in a certain *scenario* as its intrinsic property, we take inspiration from (Jaques et al., 2019) and define a new metric named *Level of Influence* (LoI). For simplicity, consider a symmetric game with two agents named Alice and Bob. Alice is the algorithm-controlled agent (*i.e.*, ego agent), and Bob can be another algorithm-controlled agent with an unknown policy or human player (*i.e.*, non-ego agent). We would like to quantify the expected impact of Bob's behavior on Alice's performance within this scenario.

Suppose Alice and Bob are algorithm-controlled agents with policy $\phi \in \Phi$ and $\theta \in \Theta$, respectively, where Φ is a policy class of size m and Θ is a policy class of size n. To account for the variations in the agents' behaviors, we assume Alice and Bob's policies are sampled from two distributions $P_\varphi(\phi) = \mathbb{P}[\varphi = \phi]$ and $P_\vartheta(\theta) = \mathbb{P}[\vartheta = \theta]$ respectively. Let $r \in \mathbb{R}$ denote the total reward Alice receives when paired with Bob. Under the two-agent game setting, Alice's reward is affected by both agents' policy choices, and the conditional reward distribution of Alice given Alice's policy $\varphi = \phi$ and Bob's policy $\vartheta = \theta$ can be represented as

$$P_{R|\vartheta,\varphi}(r|\theta,\phi) = \mathbb{P}[R = r|\vartheta = \theta, \varphi = \phi]. \tag{2}$$

We can then get Alice's marginal reward distribution as

$$P_{R|\varphi}(r|\phi) = \sum_{\theta\in\Theta} P_{R,\vartheta|\varphi}(r,\theta|\phi) = \sum_{\theta\in\Theta} P_{R|\vartheta,\varphi}(r|\theta,\phi)P_{\vartheta|\varphi}(\theta|\phi). \tag{3}$$

We propose to measure the intensity of interaction between the two agents with the discrepancy between the marginal reward distribution and the conditional reward distribution Of Alice. Intuitively, we want the LoI to measure the degree to which Alice's reward distribution changes induced by Bob's policy choice, given Alice's own policy choice. Therefore, the LoI is defined as the conditional mutual

information of Alice's reward and Bob's policy with respect to Alice's policy:

$$I(R;\vartheta|\varphi) = \mathbb{E}_{\varphi}\left[D_{\mathrm{KL}}\left(P_{R,\vartheta|\varphi}\|P_{R|\varphi}P_{\vartheta|\varphi}\right)\right] \tag{4a}$$

$$= \sum_{\phi\in\Phi} P_{\varphi}(\phi) D_{\mathrm{KL}}\left(P_{R,\vartheta|\varphi=\phi}\|P_{R|\varphi=\phi}P_{\vartheta|\varphi=\phi}\right) \tag{4b}$$

$$= \sum_{\phi\in\Phi} P_{\varphi}(\phi) \mathbb{E}_{\vartheta}\left[D_{\mathrm{KL}}\left(P_{R|\vartheta,\varphi=\phi}\|P_{R|\varphi=\phi}\right)\right] \tag{4c}$$

$$= \sum_{\phi\in\Phi} P_{\varphi}(\phi) \sum_{\theta\in\Theta} P_{\vartheta}(\theta) D_{\mathrm{KL}}\left(P_{R|\vartheta=\theta,\varphi=\phi}\|P_{R|\varphi=\phi}\right). \tag{4d}$$

It is worth noting that when $I(R;\vartheta|\varphi) = 0$, Alice's total reward will not be affected by Bob's policy choice at all under the given scenario, thus there is little value for training Alice's policy with diverse opponent's policy. The higher this value becomes, the more significant impact Bob's policy will have on Alice's expected reward; consequently, encountering a more diverse Bob's policy when training Alice's policy can help improve performance when paired with unseen partners and more training budget is well justified.

4.2 Approximation

To calculate the LoI following Equation 4d, we need to model the policy distribution of each agent, $P_{\varphi}(\phi)$ and $P_{\vartheta}(\theta)$, for a given scenario. Ideally, $P_{\vartheta}(\theta)$ should resemble the group of agents the trained agent aims to interact with. However, at the training stage, we are not aware of who the trained agent will interact with in the inference time. To this end, we propose to model $P_{\varphi}(\phi)$ and $P_{\vartheta}(\theta)$ with trained SP policies. Training a convergent SP policy gives us a pool of checkpoints, including the early-, middle-, and late-stage generations. They resemble a diverse group of agents with various skill levels and collaborating patterns, so that we can define an informative LoI metric with a small amount of computational resources. In practice, we train $a + b$ SP policies with distinct random seeds, choose a of them as Alice's policies, and the rest b policies as Bob's policies, randomly. We choose m checkpoints from the late stage of each Alice policy as a group of Alice with slightly different skills and choose n checkpoints from all stages of each Bob policy as samples of Bob's policy distribution. We summarize our LoI calculation in Algorithm 1.

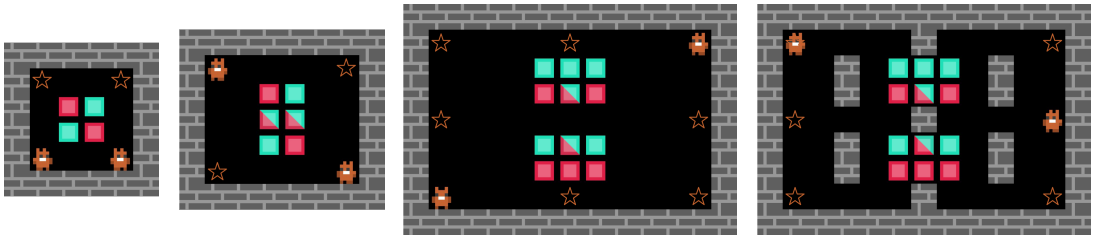
5 Environments

We adopt DeepMind's *Melting Pot* environment (Agapiou et al., 2022) for evaluation. *Melting Pot* is a MARL environment with different *substrates* (*i.e.*, physical environment) of zero-sum, shared-reward, and general-sum games. We choose the two-agent substrate named "* *in the Matrix*," whose mechanism is introduced in (Vezhnevets et al., 2019). In this game, two agents can move around the map, collect K different resources, and fire "interaction beams." Each agent has its inventory $\rho \in \mathbb{R}^K$ to track the number of resources picked up since the last re-spawn, *i.e.*, ρ_i denotes the number of the i^{th} type of resources in its inventory. The agent's inventory is only visible to itself.

An *interaction* occurs whenever one agent zaps the other agent with their interaction beam. The interaction is then resolved by a matrix game with the payoff matrix A describing the reward corresponding to the pure strategies of the matrix game available to each agent. Each kind of resource maps one-to-one to each pure strategy. During the interaction, each agent executes a mixed strategy depending on the resources they picked up before the interaction. In particular, an agent with inventory ρ plays the mixed strategy with weights $\nu = (\nu_1, \ldots, \nu_K)$, where $\nu_i = \rho_i / (\sum_{j=1}^{K} \rho_j)$. Intuitively, the more resources of a certain kind are picked up by an agent, the more likely this agent executes the corresponding strategy of that kind of resource. Formally, during the interaction, a pure strategy is sampled from each player's mixed strategy distribution defined by ν. We represent the sampled strategies of the row and column players as two one-hot vectors, denoted by $r_{\text{row}}, r_{\text{col}} \in \mathbb{R}^K$, respectively. Afterward, the rewards that the row and column players obtain from the interaction,

Chicken	*Pure Coordination*	*Prisoners Dilemma*	*Stag Hunt*
$\begin{pmatrix} 3 & 2 \\ 5 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 \\ 5 & 1 \end{pmatrix}$	$\begin{pmatrix} 4 & 0 \\ 2 & 2 \end{pmatrix}$

Table 1: Payoff matrices $\mathbf{A}_{\text{row}}$ of the two-player * *in the Matrix* game.



(a) Small (b) Medium (c) Large (d) Obstacle

Figure 1: We investigate the influence between agents under * *in the Matrix* game setting across four distinct 2-player scenarios: (a) Small, (b) Medium, (c) Large, and (d) Obstacle. Brown hollow stars denote the random spawn spots for both agents. Single-color blocks (cyan and red) denote the fixed resource, and mixed-color blocks denote the random resource, which can be changed during each initialization.

denoted by r_{row} and r_{col} respectively, are assigned via

$$r_{\text{row}} = \nu_{\text{row}}^\top A_{\text{row}} \nu_{\text{col}}, \quad r_{\text{col}} = \nu_{\text{row}}^\top A_{\text{col}} \nu_{\text{col}}. \tag{5}$$

After the interaction, both agents will re-spawn after 5 steps. Each game lasts for 2000 steps.

By changing the underlying payoff matrix, we can change the game property of the substrate. We define four different *environments* with various game properties, namely *Chicken*, *Pure Coordination*, *Prisoners Dilemma*, and *Stag Hunt* (Agapiou et al., 2022) with payoff matrices defined in Table 1. All four environments are symmetric with 2 types of resources ($K = 2$) and we have $\mathbf{A}_{\text{row}} = \mathbf{A}_{\text{col}}{}^\top$.

For each environment, we create four different *scenarios* by changing the size of the map as well as the layout of the resource and objects inside (see Figure 1). From *Small* to *Obstacle*, the map sizes are 6×6, 7×8, 9×13, and 9×13, respectively. We anticipate that different map sizes may lead to varying levels of interaction intensity among agents. The observation window of each agent is 5×5 square centered at the agent itself, which means agents in *Small* is able to observe all the resource at any spawn location.

6 Experiments Design

We design a series of experiments to validate the effectiveness of using LoI for cost-efficient generalization and demonstrate a useful application of LoI in guiding resource allocation for cost-efficient policy training. In particular, we would like to examine the following hypotheses.

Hypothesis 1. LoI is strongly correlated with the benefits of having diverse co-player distribution on the generalization of the ego agent within given scenarios (Section 6.1).

Hypothesis 2. Under the same computation budget, the set of ego agents trained with LoI-guided resource allocation can achieve higher average performance than uniform allocation (Section 6.2).

6.1 Validating the Level of Influence

As outlined in Section 4.1, our objective is to utilize LoI to predict the benefits of having diverse co-player distribution during training on the generalization. To validate this idea, we first evaluate

the impact of different co-player diversities on generalization performance within different scenarios. We then calculate the LoI of those scenarios, and find the correlation between the performance improvement and the LoI conditioned on diversity.

Fixed-Bobs Evaluation. First, we evaluate how different levels of co-player diversity impact the generalization performance within different scenarios. We train a set of policies with different levels of co-player diversity that we want to compare across all environment-scenario combinations. We then assess these trained policies against a set of predetermined agents for evaluation. Specifically, we train one SP policy for 5M steps and save a new checkpoint every 200K steps. We select four checkpoints at 1.4M, 2.6M, 3.8M, and 5M steps as a fixed group of policies for evaluation, which we refer to as "*Fixed-Bobs*".

Afterward, we train 5 instances of SP policies with different random seeds (SP), one set of PP policies with 3 populations (PP3), and one set of PP policies with 5 populations (PP5). All policies are trained for 10M steps. We choose the final checkpoint of each SP and each population of PP (*i.e.*, checkpoints at 10M steps) and pair them with the aforementioned Fixed-Bobs. Each game lasts for 2000 steps and repeats 10 times. We evaluate each training method and report the average reward for each policy by aggregating results across all 10 games, four Fixed-Bobs policies, and all populations (or all seeds in the case of SP). To better compare the performance gap between training methods across different scenarios, we normalize the previous results by dividing each element by the corresponding reward value from SP of the same environment and scenario.

LoI Calculation. Second, we estimate the LoI value for each scenario and environment following Algorithm 1. Specifically, we train 1 Alice policy ($a = 1$) and 5 Bob policies ($b = 5$) with different random seeds. We choose 4 late-stage generations at 3.8M, 4.2M, 4.6M, and 5M steps from Alice's checkpoint pool ($m = 4$) and 9 all-stage generations at 0.2M, 0.6M, 1M, 1.4M, 1.8M, 2.6M, 3.4M, 4.2M, and 5M steps from Bob's checkpoints pools ($n = 9$) of every policy. To compute the LoI, we model the policy distribution of Alice and Bob as a uniform distribution defined over their sampled checkpoints (*i.e.*, $P_\varphi = 1/4$ and $P_\vartheta = 1/9$). We perform 6 games per Alice-Bob pair ($g = 6$).

Correlation between LoI and Performance Improvement. We then calculate the average performance improvement between each training method for each environment and scenario. Suppose the average rewards for SP, PP3, and PP5 are r_1, r_2, and r_3, respectively, then we compute the average performance improvement δ as

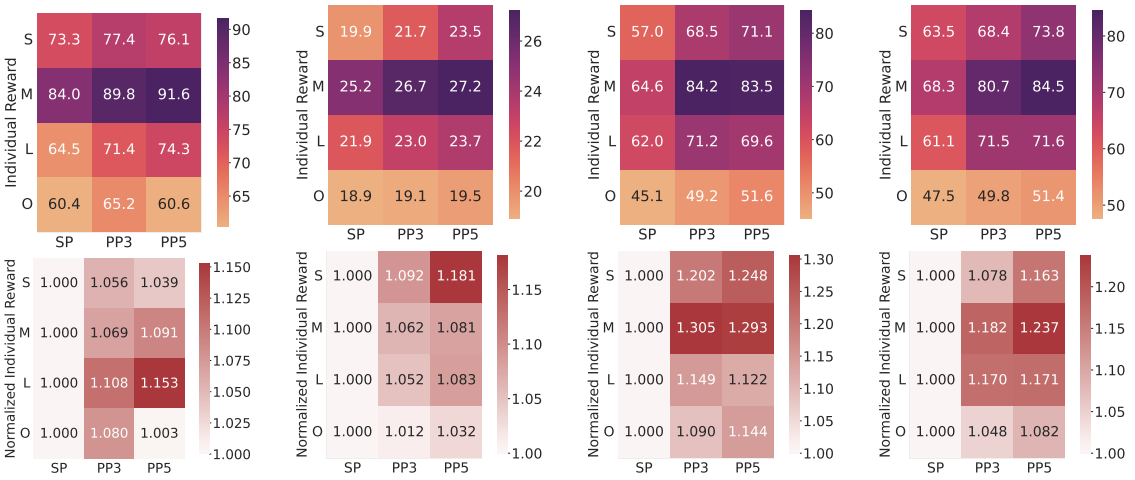
$$\delta = \frac{r_2 - r_1}{2} + \frac{r_3 - r_2}{2} = \frac{r_3 - r_1}{2}. \tag{6}$$

Last, we find the correlation between the aforementioned LoI and average improvement in four scenarios within each environment. We apply the Pearson correlation coefficient as the measurement of linear correlation. Suppose the LoIs of four scenarios on the given environment are I_i and the corresponding average improvements are δ_i. Let $\bar{I}$ and $\bar{\delta}$ denote the mean LoI and mean average improvement over four scenarios, respectively. The correlation coefficient is then calculated as

$$\gamma = \frac{\sum_{i=1}^{4}(I_i - \bar{I})(\delta_i - \bar{\delta})}{\sqrt{\sum_{i=1}^{4}(I_i - \bar{I})^2}\sqrt{\sum_{i=1}^{4}(\delta_i - \bar{\delta})^2}}. \tag{7}$$

6.2 Resource Allocation

We now show that we can utilize the proposed LoI for allocating training resource allocation under a limited computation budget. We would like to train a set of policies for a given environment (*i.e.*, a given game mechanism and reward design), with each policy tailored to a distinct scenario while keeping the total computational resource fixed. Without extra information, one may distribute resources uniformly across all scenarios. If we have access to LoI, which correlates expected performance improvement with the additional training expense, we can allocate the resource accordingly, *i.e.*, training policies with larger populations for scenarios with higher LoI and vice versa.



(a) Chicken (b) Pure Coordination (c) Prisoners Dilemma (d) Stag Hunt

Figure 2: *Top row:* Fixed-Bobs evaluation of agents trained by self-play (SP), population-play $\mathbf{p} = \mathbf{3}$ (PP3), population-play $\mathbf{p} = \mathbf{5}$ (PP5) across four scenarios: Small (S), Medium (M), Large (L), and Obstacle (O). Each tile is the average over all populations for PP and seeds for SP (5 seeds for SP), with 10 independent games between each Alice-Bob combination. *Bottom row:* Normalized individual reward of ego agents calculated by dividing each row by its first element (SP). *Result:* With a growing population, PP gains larger advantages over SP in general. However, the percentage of increase varies across different scenarios under the same environment, and the overall trend of improvement varies across different environments.

To demonstrate the proposed allocation strategy, we set a fixed training budget of 120M steps to train four policies, each handling a specific scenario. In the uniform allocation scenario, we train a 3-population PP policy (PP3) for 10M steps per seed on each scenario, summing up to 120M steps for all four scenarios. In the resource allocation approach, we calculate the LoI for each scenario (as outlined in Section 6.1) and devise a *heuristic method* to allocate resources based on this metric. By default, we allocate 30M steps for each scenario (equivalent to the cost of training PP3 for 10M steps) and compute the mean LoI across the four scenarios. Scenarios with LoI greater than one standard deviation from the mean receive 50M steps (cost of training PP5 for 10M), while those with LoI less than one standard deviation get 10M steps (cost of training SP for 10M). Adjustments are made for scenarios with LoI within one standard deviation to maintain the total budget of 120M steps. For instance, if one scenario uses only 10M steps, the saved 20M steps are reallocated to the scenario with the highest LoI among the remaining three (see Appendix D for further details). We apply the Fixed-Bobs evaluation and compare the average normalized rewards (see Section 6.1) over all four scenarios between uniform allocation and heuristic allocation.

7 Experimental Results

7.1 Validating the Level of Influence

Fixed-Bobs Evaluation. The full results of the Fixed-Bobs evaluation are shown in Figure 2. In all the environments and scenarios, ego agents' rewards demonstrate an upward trend as the training population size increases. Notably, we can regard SP as a specialized PP with a population size of 1. The absolute reward values differ significantly among diverse environments, strongly influenced by the unique game properties of each environment and the specific payoff matrix (Figure 2, top row). It suggests that our scenario design gives rise to an appropriate test bed to validate the effectiveness of the proposed LoI metric. As the maximum achievable reward varies across different scenarios, we normalize the individual reward of ego agents according to Section 6.1 to better compare the

	Chicken	*Pure Coordination*	*Prisoners Dilemma*	*Stag Hunt*
Small	1.291 (0.14)	**1.117** (0.12)	1.377 (0.11)	1.397 (0.14)
Medium	1.364 (0.09)	1.071 (0.15)	**1.385** (0.11)	**1.431** (0.13)
Large	**1.438** (0.09)	0.976 (0.09)	1.180 (0.09)	1.424 (0.07)
Obstacle	1.227 (0.17)	0.976 (0.18)	1.100 (0.12)	1.063 (0.11)

Table 2: LoI (and standard deviations, reported in parentheses) across four scenarios under four environments. Values are calculated over one Alice set ($\mathbf{m} = \mathbf{4}$) and five Bob sets ($\mathbf{n} = \mathbf{9}$) of different seeds, 10 independent games between each Alice-Bob combination. *Result:* LoI exhibits varying trends across four specified scenarios in different environments.

	Chicken	*Pure Coordination*	*Prisoners Dilemma*	*Stag Hunt*
Small	1.4130	**1.7986**	7.0535	5.1652
Medium	3.8312	1.0248	**9.4688**	**8.0993**
Large	**4.9293**	0.9117	3.7931	5.2341
Obstacle	0.0789	0.3020	3.2389	1.9517

Table 3: Average improvement on ego agents' rewards between SP, PP3, and PP5 under each scenario and environment. *Result:* The advantage of PP over SP varies across different scenarios, and the correlations between scenario and reward increment vary across different environments.

generalization performance between training methods (Figure 2, bottom row). We observe that the percentage improvement with increasing co-player diversity during training differs for each scenario within a specific environment, and the overall improvement trend varies across diverse environments.

We perform the Analysis of Variance (ANOVA) (Edwards, 2005) on the results. The ANOVA method examines whether there are significant differences in means among two or more groups. We report the F-statistic and a corresponding p-value with a null hypothesis that there is no noteworthy difference (See Appendix A Table 6). We confirm that changing the population size (*i.e.*, diversity of co-play agent's policy distribution) has a statistically significant effect on the generalization performance within different scenarios across all four environments. But most importantly, the significance of such effects varies across different scenarios for a given environment.

LoI Calculation. We calculate the mean LoI and standard deviation for each scenario and environment, as detailed in Section 6.1. The comprehensive results are presented in Table 2, with the maximum value in each environment highlighted in bold. It exhibits varying trends across four scenarios in different environments.

Correlation between LoI and Average Improvement. Based on the ego agents' rewards in Figure 2, we calculate the average improvements following Section 6.1. The results are presented in Table 3, with the highest value in each environment emphasized in bold. It is evident that the trends in each environment align closely with the LoI outcomes depicted in Table 2, which naturally sets the stage for the correlation analysis discussed in the subsequent section.

We calculate the correlation coefficient as in Section 6.1, and the results are shown in Table 4. The correlation coefficient ranges from -1 to 1. An absolute value of 1 indicates a perfect linear relationship between two groups, with all data points falling on a line. The results highlight a strong positive correlation between the average improvement of PP over SP (increasing population size) and LoI across all four environments. Consequently, we can utilize LoI as a reference to predict whether implementing a more resource-intensive training method (*e.g.*, PP with a large population size) will yield a substantial improvement in generalization over a more cost-effective training method (*e.g.*, SP or PP with a small population size) in a given scenario. Note that this correlation is valid within

	Chicken	*Pure Coordination*	*Prisoners Dilemma*	*Stag Hunt*
Statistic	0.98966	0.86309	0.93888	0.86631

Table 4: Pearson correlation coefficient between average improvement of PP over SP and LoI. *Result:* It exhibits a strong correlation between the average improvement of PP over SP and LoI under all four environments.

	Chicken	*Pure Coordination*	*Prisoners Dilemma*	*Stag Hunt*
Small	30M	50M	30M	30M
Medium	30M	30M	50M	50M
Large	50M	30M	30M	30M
Obstacle	10M	10M	10M	10M

Table 5: Allocated training resource according to the heuristic method introduced in Section 6.2. Each column adds up to 120M steps.

a specific environment. Comparing two scenarios across different environments is not meaningful in this context, since the scale of the reward varies across environments.

7.2 Resource Allocation

We utilize the LoI values provided in Table 2 for the heuristic method described in Section 6.2. The allocated training resources for each scenario are outlined in Table 5. Specifically, 10M, 30M, and 50M training steps correspond to SP, PP3, and PP5, respectively. The total steps for each environment sum up to 120M, adhering to the training budget cap. The comparison of Fixed-Bobs evaluation between uniform allocation and LoI-guided heuristic allocation is depicted in Figure 3. Notably, the heuristic allocation demonstrates a substantial improvement in the average performance across all scenarios in the Chicken, Pure Coordination, and Stag Hunt environments.
We apply the two-sample one-tailed t-test (Student, 1908) for statistical analysis. It compares the means of two independent groups and determines if one is significantly larger than the other. We provide the t-statistic and a corresponding p-value with a null hypothesis that there is no noteworthy difference (See Appendix A Table 7). We affirm that LoI-guided heuristic allocation exhibits a significant advantage over uniform allocation in all scenarios except for the Prisoners Dilemma, given the same resource budget cap. In conclusion, leveraging LoI enables us to strategically allocate resources for training a range of policies designed to handle diverse scenarios, resulting in improved overall performance within the same resource limit.

It's important to highlight that the earlier discussed heuristic allocation is based on calculating LoI using checkpoints from 1 Alice policy and 5 Bob policies, with 5M steps per policy. Consequently, employing this heuristic resource allocation necessitates an additional 30M steps beyond the 120M steps budget (25% of the total budget). In Appendix C.2, we show that, while augmenting the number of Bobs used in LoI computation significantly reduces the estimation variance, the proposed heuristic resource allocation scheme is less sensitive to the estimation noise. We can achieve comparable results as shown in Figure 3 with LoI values estimated using only 1 Alice policy and 1 Bob policy, which requires only 10M extra training steps (8.33% of the total budget). Nevertheless, we expect the estimation variance of LoI will matter when it comes to guiding resource allocation in more complex environments or other applications that require a more accurate estimation of LoI.

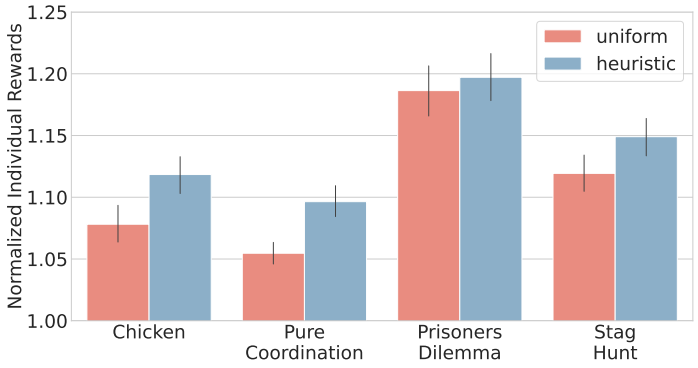


Figure 3: Fixed-Bobs evaluation comparison of the set of agents trained with uniformly allocated resource and LoI-guided heuristic allocation. Error bars correspond to **95**% confidence intervals calculated over all populations across all four scenarios with 10 independent games between each Alice-Bob combination. *Result:* The set of agents trained with LoI-guided resource allocation achieves higher overall performance under the same total resource budget in all four different environments.

8 Discussion

In our study, we introduce the *Level of Influence* (LoI) metric, a measure that quantifies the interaction intensity between agents across varied scenarios in MARL. Our proposed metric can effectively predict the potential generalization improvement by having a more diverse set of co-player policy distribution during training. Our results strongly support the strategic allocation of resources for training a tailored set of policies across diverse scenarios guided by the LoI metric. This approach consistently yields higher average performance compared to uniform allocation across different environments with distinct game properties within a limited computation budget.

Limitations and Future Work. Estimating LoI with self-play policies is susceptible to high variance. LoI essentially gauges the extent to which the policy distribution of Bob influences Alice's performance. We need a diverse set of policies (*i.e.*, diverse Bobs) to cover its potential distribution as much as possible in computing the LoI, while neither self-play nor population-play can guarantee such diversity to exist under limited training complexity. Thus, a delicate balance emerges between the expense of LoI estimation and its accuracy. Although we demonstrate that a high variance in LoI estimation does not necessarily hinder the effectiveness of the proposed resource allocation scheme, there may exist other downstream applications of LoI that require LoI estimation with substantially reduced variance. In future work, we are interested in exploring theoretical grounds and practical algorithms to generate guaranteed diverse self-play policies with minimal computation cost (Rahman et al., 2023), so that we can accurately compute LoI in a sample-efficient manner.

Moreover, while LoI serves as a valuable metric to quantify the level of interaction within a set of scenarios in the *same* environment, directly comparing the numerical values of LoI across scenarios from different environments is not meaningful. This is because the variation in reward design, which underpins the conditional mutual information calculation, affects the interpretation of these numerical values. In future work, we are interested in extending this idea in a broader context of meta-learning, where cross-environment comparisons are essential. Subsequent work may include generalizing the LoI into a comprehensive metric with predefined value ranges and thresholds across more diverse environments.

Acknowledgments

We are deeply grateful to Jiaxun Cui and Arrasy Rahman for their helpful discussions and support on this work. We also thank Micah Carroll for insightful discussion at the early stage of the project.

References

John P Agapiou, Alexander Sasha Vezhnevets, Edgar A Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, et al. Melting pot 2.0. *arXiv preprint arXiv:2211.13746*, 2022.

Noa Agmon and Peter Stone. Leading ad hoc agents in joint action settings with multiple teammates. In *AAMAS*, pp. 341–348, 2012.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *TARK*, volume 96, pp. 195–210. Citeseer, 1996.

Kalesha Bullard, Franziska Meier, Douwe Kiela, Joelle Pineau, and Jakob Foerster. Exploring zero-shot emergent communication in embodied multi-agent populations. *arXiv preprint arXiv:2010.15896*, 2020.

Kalesha Bullard, Douwe Kiela, Franziska Meier, Joelle Pineau, and Jakob Foerster. Quasi-equivalence discovery for zero-shot emergent communication. *arXiv preprint arXiv:2103.08067*, 2021.

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

Rohan Choudhury, Gokul Swamy, Dylan Hadfield-Menell, and Anca D Dragan. On the utility of model learning in hri. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 317–325. IEEE, 2019.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Anthony WF Edwards. Ra fischer, statistical methods for research workers, (1925). In *Landmark writings in western mathematics 1640-1940*, pp. 856–870. Elsevier, 2005.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Matthew C Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. On the importance of environments in human-robot coordination. *arXiv preprint arXiv:2106.10853*, 2021.

Florian Fuchs, Yunlong Song, Elia Kaufmann, Davide Scaramuzza, and Peter Dürr. Super-human performance in gran turismo sport using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 6(3):4257–4264, 2021.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *International conference on machine learning*, pp. 805–813. PMLR, 2015.

Mark K Ho, James MacGlashan, Amy Greenwald, Michael L Littman, Elizabeth Hilliard, Carl Trimbach, Stephen Brawner, Josh Tenenbaum, Max Kleiman-Weiner, and Joseph L Austerweil. Feature-based joint planning and norm learning in collaborative games. In *CogSci*, 2016.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR, 2019.

Max Kleiman-Weiner, Mark K Ho, Joseph L Austerweil, Michael L Littman, and Joshua B Tenenbaum. Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*, 2016.

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. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International conference on machine learning*, pp. 6187–6199. PMLR, 2021.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Ryan Lowe, Abhinav Gupta, Jakob Foerster, Douwe Kiela, and Joelle Pineau. On the interaction between supervision and self-play in emergent communication. *arXiv preprint arXiv:2002.01093*, 2020.

Rowan McAllister, Blake Wulfe, Jean Mercat, Logan Ellis, Sergey Levine, and Adrien Gaidon. Control-aware prediction objectives for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 01–08. IEEE, 2022.

Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes, and Joel Z Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. *arXiv preprint arXiv:2002.02325*, 2020.

Kevin R McKee, Joel Z Leibo, Charlie Beattie, and Richard Everett. Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(1):21, 2022.

Judea Pearl. Structural counterfactuals: A brief introduction. *Cognitive science*, 37(6):977–985, 2013.

Arrasy Rahman, Jiaxun Cui, and Peter Stone. Minimum coverage sets for training robust ad hoc teamwork agents. *arXiv preprint arXiv:2308.09595*, 2023.

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

Andy Shih, Arjun Sawhney, Jovana Kondic, Stefano Ermon, and Dorsa Sadigh. On the critical role of conventions in adaptive human-ai collaboration. *arXiv preprint arXiv:2104.02871*, 2021.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Peter Stone, Gal A Kaminka, and Jeffrey S Rosenschein. Leading a best-response teammate in an ad hoc team. In *International Workshop on Agent-Mediated Electronic Commerce*, pp. 132–146. Springer, 2009.

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

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

Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

Alexander Sasha Vezhnevets, Yuhuai Wu, Remi Leblond, and Joel Z Leibo. Options as responses: Grounding behavioural hierarchies in multi-agent rl. *arXiv preprint arXiv:1906.01470*, 2019.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Tom Zahavy, Vivek Veeriah, Shaobo Hou, Kevin Waugh, Matthew Lai, Edouard Leurent, Nenad Tomasev, Lisa Schut, Demis Hassabis, and Satinder Singh. Diversifying ai: Towards creative chess with alphazero. *arXiv preprint arXiv:2308.09175*, 2023.

Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. Douzero: Mastering doudizhu with self-play deep reinforcement learning, 2021.