Learning Meta Representations for Agents in Multi-Agent Reinforcement Learning

Shenao Zhang
Georgia Institute of Technology
shenao@gatech.edu

Li Shen
Tencent AI Lab
lshen.lsh@gmail.com

Lei Han
Tencent Robotics X
leihan.cs@gmail.com

Li Shen
JD Explore Academy
mathshenli@gmail.com

Abstract

In multi-agent reinforcement learning, the behaviors that agents learn in a single Markov Game (MG) are typically confined to the given agent number. Every single MG induced by varying the population may possess distinct optimal joint strategies and game-specific knowledge, which are modeled independently in modern multi-agent reinforcement learning algorithms. In this work, our focus is on creating agents that can generalize across population-varying MGs. Instead of learning a unimodal policy, each agent learns a policy set comprising effective strategies across a variety of games. To achieve this, we propose *Meta Representations for Agents* (MRA) that explicitly models the game-common and game-specific strategic knowledge. By representing the policy sets with multi-modal latent policies, the game-common strategic knowledge and diverse strategic modes are discovered through an iterative optimization procedure. We prove that by approximately maximizing the resulting constrained mutual information objective, the policies can reach Nash Equilibrium in every evaluation MG when the latent space is sufficiently large. When deploying MRA in practical settings with limited latent space sizes, fast adaptation can be achieved by leveraging the first-order gradient information. Extensive experiments demonstrate the effectiveness of MRA in improving training performance and generalization ability in challenging evaluation games.

1 Introduction

Behaviors of agents learned in a single Markov Game (MG) highly depend on the environmental settings, especially the number of agents, i.e., population (Suarez et al., 2019; Long et al., 2020). Many multi-agent reinforcement learning (MARL) algorithms (Sukhbaatar et al., 2016; Foerster et al., 2016; Lowe et al., 2017) are developed in games with a fixed population. However, the algorithms may suffer from generalization issues, i.e., the policies learned in a single MG are brittle to the change of the agent number (Suarez et al., 2019). Recent works have experimentally shown the benefit of knowledge transfer between MGs with different populations (Agarwal et al., 2019; Long et al., 2020), which is required to perform between successive games. Unfortunately, the resulting agents are still confined to particular training games, with less ability for extrapolation.

In this work, we are concerned with learning multi-agent policies that generalize across Markov Games constructed by varying the population from the same underlying environment. The created agents are expected to behave well in both training MGs and novel (or unseen) evaluation MGs. However, for each agent, optimizing one unimodal policy even to maximize the performance on the entire *training* MG set is still challenging (Teh et al., 2017). Effective policies in population-varying games, such as ones that achieve Nash Equilibrium in each game, may behave dramatically different due to the game-specific strategic knowledge of themselves. Such discrepancy will hamper the performance in individual games (Brunskill & Li, 2013). In this regard, it is desirable to learn *sets of policies* that are formed by the optimal strategies for each training MG, while transferring knowledge to *unseen* MGs is still challenging nevertheless.

To cope with this generalization challenge, our solution involves modeling the *game-specific* and *game-common* strategic knowledge. In unseen games, although the optimal game-specific knowledge that leads to optimal policies is unobtainable, the game-common knowledge and various strategic modes can be captured during training by imposing *knowledge variations*, i.e., the *suboptimal* game-specific knowledge. Instead of only fitting the best response, learning to make decisions under multiple knowledge variations plays the role of augmenting the training games. As a result, the strategic knowledge is learned in an unsupervised manner and agents can effectively generalize to novel MGs.

For games induced by varying the population, the distinct optimal policies are determined by different strategic relationships between agents (see Equation 1 for a formal definition), which we characterize as game-specific knowledge. For example, in a Pac-Man game that is populationally dominated by ghost agents, the game-specific knowledge for Pac-Man is to focus on the ghost agents' positions for survival. However, as all Pac-Man agents ignore other Pac-Man agents' positions, they are unaware of collaborating and their ability to collect food dots in evaluation games is poor. One potential fix is to learn the representations which can serve as the game-common knowledge and generate a set of policies that output the best-effort actions when conditioned on different (suboptimal) strategic relationships (i.e., knowledge variations) between agents, e.g., by forcing the Pac-Man to pay more attention to other Pac-Man in spite of the game being ghost-dominated. Although these policies can be suboptimal in the training MG, they have the potential to perform well in evaluation games in a zero-shot manner. This motivates us to learn *diverse* strategies so that even if not all knowledge variations are covered during training, the game-common knowledge can be learned. Better generalization is thus achieved since we only need to fit the best strategic relationship in the evaluation game.

To formalize this intuition, we propose *Meta Representations for Agents* (MRA) to discover the underlying strategic structures. Specifically, by meta-representing the policy sets with multi-modal latent policies, the game-common knowledge and diverse strategic modes are captured through iterative diversity-driven optimization. We prove that by approximately maximizing a constrained mutual information objective, the latent policies can reach Nash Equilibrium in every evaluation game if with a sufficiently large latent space. When deployed in limited-size latent spaces, fast adaptation is achieved by leveraging the first-order gradient information. We further empirically validate the benefits of MRA, which is capable of boosting training performance and extrapolating over a variety of evaluation games.

2 Related Work

Multi-Agent Reinforcement Learning (MARL) extends RL to multi-agent systems. In this work, we follow the centralized training with decentralized execution (CTDE) setting. Recent MARL algorithms with CTDE setting (Lowe et al., 2017; Jiang & Lu, 2018; Iqbal & Sha, 2018) learn in MGs with fixed numbers of agents. However, the resulting agents are shown to be unable to play well in some situations. For instance, it is shown in (Suarez et al., 2019) that the random exploration bottleneck may result in both inferior and brittle policies, e.g., competitive agents trained in a small population lack adequate exploration compared with agents trained in a large population. On the other hand, the complexity of games grows exponentially with the population, which causes direct learning intractable (Yang et al., 2018). For both the above difficulties, training in multiple MDPs or MGs is shown to be helpful (Teh et al., 2017; Wang et al., 2020b; Long et al., 2020), which highlights the importance of learning transferable knowledge.

In single-agent RL, knowledge transfer is proven useful to improve the performance in related MDPs (Taylor & Stone, 2009), e.g., knowledge from game-specific experts distilled to a policy (Rusu et al., 2015; Parisotto et al., 2015; Teh et al., 2017). Recently, generalizable policy sets are learned in SMERL (Kumar et al., 2020), which share similarities with our work. However, SMERL is proposed to improve the single-agent policy robustness, with the motivation that *remembering* diverse (suboptimal) policies in a *single* MDP can directly lead to robust behaviors, with no need to perform explicit perturbations. Similar ideas to learn transferable skills also appear in recent works (Lim et al., 2021; Xie et al., 2021). Approaches that also adopt latent variable policies and mutual information objectives (Eysenbach et al., 2019; Sharma et al., 2020; Mahajan et al., 2019; Zheng & Yue, 2018) differ from ours since they either focus on the unsupervised skill discovery in single MDPs or learn in multi-task setups without generalization guarantees.

Meta-learning for RL (Vilalta & Drissi, 2002), including Reptile (Nichol et al., 2018) and RL^2 (Duan et al., 2016), is related to our work, which extracts the *prior knowledge* in related MDPs by e.g., recurrent models (Wang et al., 2016; Duan et al., 2016) or feed-forward models (Brunskill & Li, 2013). Alternatively, the gradient information can also be leveraged to meta-learn (Finn et al., 2017; Nichol et al., 2018). However, in our work, the learning protocol is more like multi-task learning in the transfer learning literature. The common knowledge in MRA is different from the prior knowledge in meta-learning as the latter captures the high-level essence, while the former is the feature that directly transfers. Also, MRA by *explicitly* modeling the common knowledge and specific knowledge is more suitable for specific problems and has more interpretability.

Recent MARL works (Agarwal et al., 2019; Wang et al., 2020b; Long et al., 2020) experimentally reveal the performance benefits of transferring knowledge between population-varying MGs, but with the ultimate goal of training in complex large-population MGs, achieved by curriculum learning. There are also works focusing on multi-task MARL, e.g., (Omidshafiei et al., 2017) with independent learners. Although transfer learning is more about the intra-agent transfer, i.e., between MGs, the inter-agent transfer is also addressed by parameter sharing between cooperative agents (Tan, 1993; Terry et al., 2020) or between homogeneous agents in role-symmetric games (Suarez et al., 2019; Muller et al., 2020). With the aim of learning dynamic team composition of *heterogeneous* agents, COPA (Liu et al., 2021a) introduced a coach-player framework in *single* training games. Besides, the counterfactual reasoning in REFIL (Iqbal et al., 2021) focused on the multi-task training setting, while we study the generalizability of MARL agents to unseen evaluation games with a theoretically justified objective. There are also works that aim to discover diverse strategic behaviors in MARL (Tang et al., 2021; Lupu et al., 2021), with randomized policy gradient and zero-shot coordination, respectively. Notably, approaches that model the *dynamical* interaction between agents (Wang et al., 2019; Yang et al., 2021) are orthogonal to the *strategic* modeling in our work.

3 Preliminaries

Markov Game: An N-agent Markov Game m is defined by the state space $\mathcal{S}$, action sets $\{\mathcal{A}^1, \dots, \mathcal{A}^N\}$, and observation sets $\{\mathcal{O}^1, \dots, \mathcal{O}^N\}$. For any agent $i \in [1, N]$, $o^i \in \mathcal{O}^i$ is an observation of the global state $s \in \mathcal{S}$. The state transition and the reward function for agent i are defined as $\mathcal{P}_m : \mathcal{S} \times \mathcal{A}^1 \times \ldots \times \mathcal{A}^N \to \Delta(\mathcal{S})$ and $\mathcal{R}^i_m : \mathcal{S} \times \mathcal{A}^i \to [0, 1]$, respectively, where $\Delta(\mathcal{S})$ denotes the set of discrete probability distributions over $\mathcal{S}$. The joint strategy is denoted as $\boldsymbol{\pi} = (\pi^1, \dots, \pi^N) = (\pi^i, \boldsymbol{\pi^{\text{-}i}})$, where π^i is the strategy of agent i and $\boldsymbol{\pi^{\text{-}i}}$ is the joint strategy excluding it. In the following sections, we study the Markov Games with discrete state and action spaces.

In this work, we consider role-symmetric MGs (Suarez et al., 2019; Muller et al., 2020), where homogeneous agents are with the same reward function and action space. The number of types of homogeneous agents is denoted as h, e.g., $h = 2$ in an N-agent Pac-Man game representing Pac-Man and ghost agents. Homogeneous agents are symmetric in each game, i.e., changing the policy of an agent with another homogeneous agent will not affect the outcome (Terry et al., 2020). Population-varying MGs, including training and evaluation MGs, e.g., varying N_1 and N_2 in an N_1 Pac-Man, N_2 ghosts game, are with the same h and with states from state set S, whereas described by different transition $\mathcal{P}$ and joint space of observation $\mathcal{O}$, action $\mathcal{A}$, reward $\mathcal{R}$.

Relational Representation: As an opponent modeling framework, relational representation (Long et al., 2020; Agarwal et al., 2019; Iqbal & Sha, 2018; Zhang et al., 2021) aims to capture the strategic relationship between agents and output an embedding e for further policy and critic function learning. Specifically, consider the observation o^i of agent i with entities $o^i = \left[o^i_s, o^i_1, \dots, o^i_j, \dots, o^i_N\right]$, where o^i_s is agent i's self properties (e.g., its speed), o^i_j is agent i's observation on agent j (e.g., distance from agent j), and the observed environment information (e.g., landmark locations) is concatenated to these entities. Then with self-attention (Vaswani et al., 2017) generating the pair-wise relation $g^{i,j}$, i.e., the j-th entity of agent i's (egocentric) relational graph g^i, the representation embedding e^i for agent i is formulated as

$$e^i = \sum_{j \neq i} g^{i,j} \mathcal{V}(o^i_j), \text{where } g^{i,j} = \frac{\exp(\mathcal{Q}(o^i_s)^\top \mathcal{K}(o^i_j))}{\sum_{j \neq i} \exp(\mathcal{Q}(o^i_s)^\top K(o^i_j))}, \qquad (1)$$

where we follow the Transformer architecture (Vaswani et al., 2017) and let $\mathcal{V}(\cdot)$, $\mathcal{Q}(\cdot)$ and $\mathcal{K}(\cdot)$ represent linear functions. The observation embedding with an arbitrary number of agents can thus be represented with a fixed length.

Nash Equilibrium: A core concept in game theory is Nash Equilibrium (NE). When every agent in the MG m acts according to the joint strategy $\boldsymbol{\pi}$ at state s, the value of agent i, denoted by $v^{i,m}_{\boldsymbol{\pi}}(s)$, is the expectation of i's γ-discounted cumulative reward. Formally, we define

$$v^{i,m}_{\boldsymbol{\pi}}(s) = \mathop{\mathbb{E}}_{\boldsymbol{a} \sim \boldsymbol{\pi}, s_0 = s, s_t \sim \mathcal{P}_m} \left[\sum_t \gamma^t r^i_m(s_t, \boldsymbol{a_t}) \right].$$

In this work, the bold symbol is joint over all agents, and variables with superscript i are of agent i. Denote the value of the best response for agent i as $v^{*i,m}_{\boldsymbol{\pi}^{\text{-}i}}$, which is the best policy of agent i when $\boldsymbol{\pi^{\text{-}i}}$ is executed, i.e., $v^{*i,m}_{\boldsymbol{\pi}^{\text{-}i}} = \max_{\pi^i} v^{i,m}_{\pi^i, \boldsymbol{\pi}^{\text{-}i}}$. Then the joint strategy $\boldsymbol{\pi}$ reaches NE if for any agent $i \in \{1, ..., N\}$, $v^{i,m}_{\boldsymbol{\pi}}(s) = v^{*i,m}_{\boldsymbol{\pi}^{\text{-}i}}(s)$.

A common metric to measure the distance to a Nash Equilibrium is NashConv, which represents how much each player (or agent) gains by deviating from the best response (unilaterally) in total. And it can be approximately calculated in small games (Johanson et al., 2011; Lanctot et al., 2017). We denote the NashConv of $\boldsymbol{\pi}$ in the Markov Game m as $\mathcal{D}_m(\boldsymbol{\pi})$, defined as

$$\mathcal{D}_m(\boldsymbol{\pi}) = \mathcal{D}_m(\pi^i, \boldsymbol{\pi}^{\text{-}i}) = \left\| \left\| v^{*i,m}_{\boldsymbol{\pi}^{\text{-}i}} - v^{i,m}_{\boldsymbol{\pi}} \right\|_{s,\infty} \right\|_{i,1} = \sum_{1 \le i \le N} \max_{s \in \mathcal{S}} |v^{*i,m}_{\boldsymbol{\pi}^{\text{-}i}}(s) - v^{i,m}_{\boldsymbol{\pi}}(s)|,$$

where $\|\cdot\|_{s,\infty}$ is the $\mathcal{L}_{+\infty}$-norm over the state space $\mathcal{S}$ and $\|\cdot\|_{i,1}$ is the $\mathcal{L}_1$-norm over agent indexes. With this definition, the joint strategy $\boldsymbol{\pi}$ reaches NE in m if and only if $\mathcal{D}_m(\boldsymbol{\pi}) = 0$.

4 Learning Meta Representations for Agents

4.1 Problem Statement

In a single Markov Game, achieving Nash Equilibrium gives reasonable solutions and is of great importance (Hu & Wellman, 2003; Yang et al., 2018; Pérolat et al., 2017). To enable generalization in different MGs, the most straightforward way is to learn a joint strategy set $\mathbf{\Pi}$ that contains effective joint strategies for every MG, e.g., the ones that achieve NE. We denote the set of all training MGs as $\mathcal{M}$ and the set of evaluation MGs as $\mathcal{M}'$. Then the goal is to learn an optimal joint strategy set $\mathbf{\Pi}^*$ that satisfies

$$\forall m' \in \mathcal{M}', \exists \boldsymbol{\pi} \in \mathbf{\Pi}^*, \text{ s.t. } \mathcal{D}_{m'}(\boldsymbol{\pi}) = 0. \tag{2}$$

Consider the problem of optimizing $\mathbf{\Pi}$ so that the optimal $\mathbf{\Pi}^*$ satisfies Equation 2. We first need its size $|\mathbf{\Pi}|$ to be sufficiently large to comprise at least one effective strategy for every $m' \in \mathcal{M}'$. Then $\mathbf{\Pi}$ should be improved with respect to the worst-performing m', i.e., the game with no effective strategy contained in $\mathbf{\Pi}$, to achieve low regret $\mathcal{D}_{m'}$ for all m'. In other words, $\mathbf{\Pi}$ is updated to include the joint strategy $\boldsymbol{\pi}$ that minimizes $\mathcal{D}_{m'}(\boldsymbol{\pi})$. Formally,

$$\mathbf{\Pi}^* = \arg\min_{\mathbf{\Pi}} \mathcal{L}(\mathbf{\Pi}), \text{ where } \mathcal{L}(\mathbf{\Pi}) = \min_{\boldsymbol{\pi}\sim\mathbf{\Pi}} \max_{m'\in\mathcal{M}'} \mathcal{D}_{m'}(\boldsymbol{\pi}). \tag{3}$$

However, minimizing $\mathcal{L}(\mathbf{\Pi})$ over the unseen evaluation games in $\mathcal{M}'$ is impractical in general. In the following sections, we cope with this intractability by introducing a heuristic algorithm and showing that the resulting objective is indeed equivalent to Equation 3, optimizing which can lead to the optimal strategy set that satisfies Equation 2.

4.2 Relational Representation with Latent Variable Policies

Instead of learning independent unimodal policies to form the set $\mathbf{\Pi}$, we adopt hierarchical latent variable policies to represent the multimodality, with the game-common and game-specific strategic knowledge explicitly modeled by relational representation. Specifically, for population-varying MGs, we regard the relational graph g as the game-specific knowledge since (1) agents optimally behave in each game by learning the *per-game optimal* relational graph; and (2) agents take different actions when incorporating different strategic relationships so that multiple strategic modes are obtained with varied g. For this reason, we treat g as a high-level latent variable that is dynamically generated by $g = \phi(o, z)$, where z is a low-level latent sampled from a learned distribution $p(z \mid m; \psi)$. An agent takes action $a \sim \pi(\cdot \mid o, g; \theta)$, where $g = \phi(o, z)$ and $z \sim p(z \mid m; \psi)$. Then the common knowledge that determines how agents can optimally behave conditioned on different g is distilled into the policy parameter θ, which includes the transformation $\mathcal{V}$ in Equation 1 and the successive policy network parameters.

Our implementation is illustrated in Figure 1. Specifically, different relational graphs are controlled by the lower-level latent variable z, which selects the attention heads inspired by the *option* (Sutton et al., 1999) structure. We apply the multi-head self-attention architecture (Vaswani et al., 2017) and set the head number as the dimension of the categorical distribution $p(z)$. Then with a sampled latent, say z_1, the relational graph g_1 corresponds to the output at the z_1-th head. The same relational graph is used for both the decentralized actor and the centralized critic.

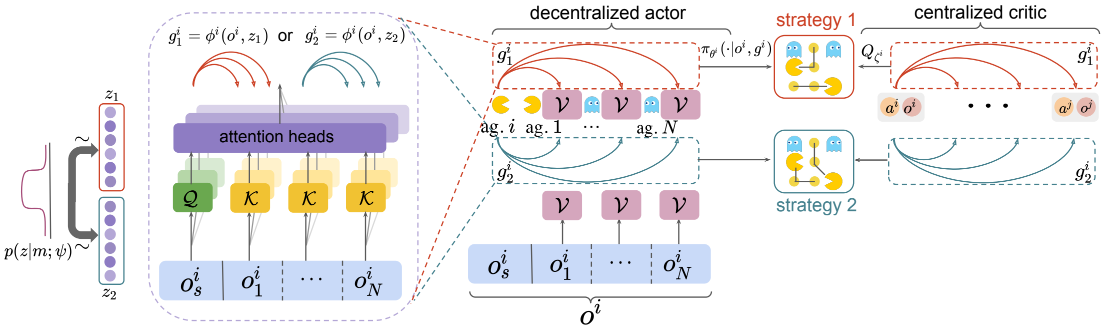


Figure 1: The illustration of the implemented architecture of our MRA algorithm. The observation o^i of agent (ag.) i contains the agent's self-information o_s^i and the observed information of other agents, i.e., $o_1^i, \ldots, o_N^i$. After sampling the latent z and computing the (egocentric) relational graph g^i, the action and estimated value of agent i can be generated. Here, we show how different strategy modes can be generated, which correspond to z_1, g_1^i and z_2, g_2^i.

Unfortunately, although the above design fits well into the multi-task MARL setup, where each task differs in the number of agents, there is no guarantee that the resulting agents can generalize to novel evaluation MGs. To address this issue, we present the insights and objectives of our *Meta Representations for Agents* (MRA) algorithm as follows.

4.3 Generalization by Strategic Knowledge Discovery

Our core idea to enable generalization is to discover the underlying strategic structures in the underlying games. Although the effective policies in the evaluation games are never known during training, agents can still learn the common strategic knowledge and different behavioral modes solely in the *training* games in an unsupervised manner with the imposed *suboptimal* game-specific knowledge. In particular for population varying games, the agents are assigned different strategic relationships, i.e., each agent pays additional attention to some agents while ignoring others. Instead of learning only one optimal joint policy, training with multiple strategic relationships enables the unsupervised discovery of behavior modes, some of which offer appreciable returns in evaluation MGs. Thus, when evaluating in novel games, the desired policy behaviors can be quickly generated by adaptation. In the extreme case that sufficiently many strategic modes are captured with an extremely large latent space, the desired policy for evaluation games can be directly found.

Specifically, agents optimally behave in each game $m \in \mathcal{M}$ with the optimal policy parameter θ^* and the (per-game) optimal relational graph g^*. By imposing *knowledge* (or *relation*) *variations* in m, i.e., *multiple suboptimal* g at a certain observation, agents learn how the best decisions to accomplish the task are made, i.e., learn θ^* that achieves the highest average return. With the discovery of distinct strategic modes, the game-common knowledge contained in θ^* is obtained. Thus, when agents are in the novel MGs $m' \in \mathcal{M}'$, their policies can effectively adapt by learning the optimal relational graph in m', or achieve zero-shot transfer (without adaptation) if the latent space is large. This gives the objective of θ^i that maximizes the average return of all knowledge variations and all training MGs:

$$\max_{\theta^i} \mathcal{L}(\theta^i) = \max_{\theta^i} \mathop{\mathbb{E}}_{m\sim\mathcal{M}, g^i} \left[v^{i,m}_{\pi^i(\cdot|\cdot,g^i;\theta^i),\boldsymbol{\pi}^{\text{-}i}} \right]. \tag{4}$$

Notably, Equation 4 differs from the objective of multi-task learning where the average training return is maximized by learning the optimal θ^* and a *single optimal* relational graph in each game.

In order to perform well in all $m' \in \mathcal{M}'$, the strategic modes captured during training should cover as many behaviors as possible. This requires a large latent space size $|Z|$ and *diverse* actions. Therefore, we introduce a diversity-driven objective that encourages high mutual information between g and a for behavior diversity, as well as between m and g to extract game-specific knowledge, defined as follows:

$$\max_{\psi^i,\phi^i} \mathcal{L}(\psi^i, \phi^i) = \max_{\psi^i,\phi^i} \mathcal{I}(g^i; a^i \mid o^i) + \mathcal{I}(m; g^i \mid o^i), \tag{5}$$

where $\mathcal{I}(g; a \mid o) = \mathcal{H}(a \mid o) - \mathcal{H}(a \mid o, g)$ is the mutual information between g and a conditioned on observation o. With a slight abuse of notation, the variable m denotes the basic information of game m, such as the numbers of agents of each set of homogeneous agents.

4.4 Fast Adaptation with Limited Latent Space Size

Despite the diversity-inducing objective, we also need a large enough latent space $|Z|$. Specifically, denote by $\boldsymbol{\Pi}_\Theta$ the joint strategy set parameterized by $\Theta = \{\psi, \phi, \theta\}$. Then achieving zero-shot adaptation requires $|Z| = |\boldsymbol{\Pi}_\Theta| \geq |\boldsymbol{\Pi}^*|$. However, without further assumptions on the Markov Games in the evaluation set $\mathcal{M}'$, the size $|\boldsymbol{\Pi}^*|$ will be unbounded. Therefore, with practical limited-size latent models, we adopt the techniques from Reptile (Nichol et al., 2018) to fast adapt to evaluation games, achieved by repeatedly selecting a game and moving the parameter towards the trained weights on this game to find the point near all games' solution manifolds.

Specifically, we optimize θ by performing K policy gradient steps on each individual MG, instead of trying to maximize the average return over all training games in a joint way. Formally, after selecting a training MG m, the objective for θ in the k-th mini-batch changes from Equation 4 to $\mathcal{L}^k_m(\theta^i) = \mathbb{E}_{g^i}[v^{i,m}_{\pi^i(\cdot|\cdot,g^i;\theta^i),\boldsymbol{\pi}^{\text{-}i}}]$. Let $U^K_m(\theta)$ denote the policy parameter after K gradient steps[1] with learning rate β. Then θ is updated by $\theta \leftarrow \theta + \alpha\Delta\theta$, where α is a hyperparameter and $\Delta\theta = U^K_m(\theta) - \theta$. By doing so, the first-order gradient information can be leveraged to update θ towards the instance-specific adapted policy parameter. This can be seen by writing the expected update $\mathbb{E}[\Delta\theta]$ over mini-batches in m as

$$\mathbb{E}[\Delta\theta] = (K-1)\mathbb{E}\left[\nabla\mathcal{L}^k_m(\theta)\right] + \frac{(K-1)(K-2)\beta}{2}\mathbb{E}\left[\nabla\left(\nabla\mathcal{L}^k_m(\theta)\nabla\mathcal{L}^j_m(\theta)\right)\right], \tag{6}$$

[1] The gradient can be calculated by leveraging a centralized value estimation as described in Equation 9.

where $\nabla\mathcal{L}_m^k(\theta)$ is the gradient at the initial θ. The derivation is in Appendix B. Notably, the second term on the RHS in Equation 6 is with the direction that increases the inner product between gradients of different mini-batches j, k. That is, θ is optimized not only to maximize the return under all relation variations, but also towards the place where gradients of different variations point to the same direction, i.e., the place that is easy to optimize from. With this property, when the optimal strategic modes of evaluation MGs are not discovered during training, the learned θ can still fast adapt to effective policies.

The pseudocode of *Meta Representations for Agents* (MRA) is presented in Algorithm 1, where iterative optimization of Equation 4 and Equation 5 is performed to update the policy parameter θ and the parameters ψ, ϕ in the relational representation. It is worth noting that the right pseudocode is a high-level abstract of our method. We will discuss the optimization procedure and the algorithm implementation in Section 6 and Appendix D.

Algorithm 1 MRA: Training in the MG set $\mathcal{M}$

```
while not converged do
    for MG $m \in \mathcal{M}$ with population $N_m$ do
        Every agent samples $z^i \sim p_{\psi^i}(\cdot|m)$ and $g^i = \phi^i(o^i, z^i)$
        Execute $a^1, \dots, a^{N_m}$ in $m$, where $a^i \sim \pi_{\theta^i}(\cdot|o^i, g^i)$
        for agent $i = 1, \dots, N_m$ do
            Update $\theta^i$ by $\theta^i \leftarrow \theta^i + \alpha(U_m^K(\theta^i) - \theta^i)$
            Update $\psi^i$ and $\phi^i$ by Equation 5
        end for
    end for
end while
```

5 Analysis

In this section, we provide a theoretical analysis of MRA. Specifically, we show that the optimal parameters resulting from the MRA objective in Equation 4 and Equation 5 ensure generalizability under certain conditions. To begin with, we introduce the Markov state transition operator $\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{\text{-}i}}$ in MG m, defined as

$$\left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{\text{-}i}} x\right)(s) = \int_{s'\sim\mathcal{S}} x(s') \mathop{\mathbb{E}}_{\pi^i,\boldsymbol{\pi}^{\text{-}i}} \left[\mathcal{P}_m(ds' \mid s, a^i, \boldsymbol{a}^{\text{-}i})\right].$$

Here, $x\colon \mathcal{S} \to \mathbb{R}$ is an L_1 Lebesgue integrable function. The norm of the operator is defined as $\|\Lambda\|_{op} := \sup\{\|\Lambda x\|_{s,1}\colon \|x\|_{s,1} \le 1\}$.

Then we make the assumption of Lipschitz Game.

Assumption 1. *(Lipschitz Game). For any Markov Game $m \in (\mathcal{M} \cup \mathcal{M}')$, there exists a Lipschitz coefficient $\iota_m > 0$ such that for all agent in m and $s \in \mathcal{S}$:*

$$\left\|\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{\text{-}i}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{\text{-}i}}\right\|_{op} \le \iota_m \left\| \left\|\pi^{*i}(a\mid s) - \pi^i(a\mid s)\right\|_{a,1}\right\|_{s,\infty},$$

where $\|\cdot\|_{a,1}$ is the $\mathcal{L}_1$-norm over the action space $\mathcal{A}$.

The Lipschitz assumption has been made in a plethora of preceding studies (Liu et al., 2021b; Zhang et al., 2019; Zhang, 2022). We note that Assumption 1 is reasonable since the Lipschitz coefficient ι_m can be interpreted as the *influence* of agents (Radanovic et al., 2019; Dimitrakakis et al., 2019), which measures how much the policy changing of an agent can affect the game environment.

Then we define a distance metric that measures the discrepancy between $\mathcal{M}$ and $\mathcal{M}'$ by comparing and computing the distance to NE in the games of the two sets. Let N_m denote the total number of agents in game m, and $h_{i,m}$ denote the homogeneous agent set of agent i in m.

Definition 1. *For two sets of MGs $\mathcal{M}$ and $\mathcal{M}'$, we define the distance ς between $\mathcal{M}$ and $\mathcal{M}'$ by*

$$\varsigma = \max_{\substack{m'\in\mathcal{M}'\\ i\in\{1,\dots,N_{m'}\}}} \min_{\substack{m\in\mathcal{M}, i'\in h_{i,m}\\ \boldsymbol{\pi}\in\{\boldsymbol{\pi}|\mathcal{D}_m(\boldsymbol{\pi})=0\}\\ \boldsymbol{\pi}'\in\{\boldsymbol{\pi}'|\mathcal{D}_{m'}(\boldsymbol{\pi}')=0\}}} \mathcal{D}_{m'}(\pi^{i'}, \boldsymbol{\pi}'^{\text{-}i}).$$

We also define the ϵ-range joint strategy set $\hat{\boldsymbol{\Pi}}$ to guide the policy learning of agents during training.

Definition 2. *For the training MG set $\mathcal{M}$ and $\epsilon > 0$, the ϵ-range joint strategy set $\hat{\boldsymbol{\Pi}}$ is defined as:*

$$\hat{\boldsymbol{\Pi}} = \bigcup_{m\in\mathcal{M}} \hat{\boldsymbol{\Pi}}_{\boldsymbol{m}}, \quad \textit{where} \quad \hat{\boldsymbol{\Pi}}_{\boldsymbol{m}} = \{\boldsymbol{\pi} \mid \mathcal{D}_m(\boldsymbol{\pi}) \le \epsilon\}.$$

By bounding ϵ that characterizes a large set $\hat{\boldsymbol{\Pi}}$, we show by the following theorem that Equation 3 can be solved from the constrained mutual information maximization objective.

Theorem 1. *If* $|\mathbf{\Pi}_\Theta| \geq |\hat{\mathbf{\Pi}}|$ *and* ϵ *satisfies* $\epsilon \geq \varsigma - \min_{\iota_m, \iota_{m'}} \frac{\varsigma\gamma(\iota_{m'} - \iota_m)}{\gamma\iota_{m'}+1-\gamma}$, *then with the optimal parameters* $\Theta^* = \{\psi^*, \phi^*, \theta^*\}$ *given by*

$$\psi^*, \phi^* = \underset{\psi,\phi}{\arg\max}\, \mathcal{I}(g; a \mid o) + \mathcal{I}(m; g \mid o) \;\; s.t. \;\; \boldsymbol{\pi}_{\theta^*} \in \hat{\mathbf{\Pi}}, \tag{7}$$

for every evaluation Markov Game $m' \in \mathcal{M}'$, *there exists a joint strategy* $\boldsymbol{\pi} \in \mathbf{\Pi}_{\Theta^*}$ *that reaches Nash Equilibrium (i.e.,* $\mathbf{\Pi}_{\Theta^*} = \mathbf{\Pi}^*$ *satisfies Equation 2). Here, the variables and parameters are per-agent, e.g.,* $\boldsymbol{\pi}_\theta$ *is joint over* π_{θ^i}*, and the superscript is omitted for clarity.*

Proof. See Appendix A for a full proof and Appendix A.1 for a proof sketch. □

Theorem 1 suggests a general paradigm of diversity-driven learning that is effective when $\hat{\mathbf{\Pi}}$ satisfies certain properties. In practical MGs, however, the unknown Lipschitz coefficient and the hardness of calculating ς pose challenges to computing the satisfying ϵ. An approximation to the optimal parameters in Equation 7 is to perform iterative optimization following Equation 4 and Equation 5.

With fixed ϕ and ψ, the objective of θ in Equation 4 (greedily) maximizes the expected value over variations in order to minimize the distances to Nash of different policy modes. In other words, the distance $\mathcal{D}_m(\boldsymbol{\pi})$ to the equilibrium of the corresponding joint strategies is minimized to satisfy $\mathcal{D}_m(\boldsymbol{\pi}) \leq \epsilon$ in the long run. Then the optimization of ϕ and ψ follows to maximize $\mathcal{I}(g; a \mid o) + \mathcal{I}(m; g \mid o)$. By iteratively improving the mutual information and updating θ towards the ϵ-range $\hat{\mathbf{\Pi}}$, the obtained solutions are close to the optimal parameters in Equation 7. Besides, the condition $|\mathbf{\Pi}_\Theta| \geq |\hat{\mathbf{\Pi}}|$ in the theorem supports the intuition that a sufficiently large policy set (or latent space) is required for zero-shot transfer. However, as an approximation to the theorem, MRA also has some limitations, which we discuss and provide potential improvements in Section 8.

6 Objective Optimization

We have shown that the iterative optimization of MRA arises from a theoretically justified objective. In this section, we present the optimization procedures in Line 6 and Line 7 of Algorithm 1 that correspond to the policy gradient and the mutual information maximization, respectively.

6.1 Multi-Agent Actor-Critic Policy Gradient

The policy parameter θ^i in objective Equation 4 is optimized by introducing a *centralized* critic Q_{ζ^i} for each agent i (Lowe et al., 2017). Denote the target network with delayed policy and critic parameters as $\bar{\theta}$, $\bar{\zeta}$, and replay buffer as D. The parameterized critic Q_{ζ^i} is optimized to minimize

$$\mathcal{L}(\zeta^i) = \mathop{\mathbb{E}}_{(\boldsymbol{o},\boldsymbol{a},\boldsymbol{o'},\boldsymbol{r})\sim D}\left[\left(Q_{\zeta^i}(\boldsymbol{o},\boldsymbol{a}) - y^i\right)^2\right], \text{where } y^i = r^i + \gamma \mathop{\mathbb{E}}_{\boldsymbol{a'}\sim\boldsymbol{\pi}_{\bar{\theta}}}\left[Q_{\bar{\zeta}^i}(\boldsymbol{o'},\boldsymbol{a'})\right] \tag{8}$$

Then the gradient of the policy parameter θ^i of agent i during training is given by

$$\nabla_{\theta^i}\mathcal{L}(\boldsymbol{\pi}) = \mathop{\mathbb{E}}_{(\boldsymbol{o},\boldsymbol{g})\sim D, \boldsymbol{a}\sim\boldsymbol{\pi}}\left[\nabla_{\theta^i}\log\pi_{\theta^i}(a^i \mid o^i, g^i) Q_{\zeta^i}(\boldsymbol{o},\boldsymbol{a})\right], \tag{9}$$

where $Q_{\zeta^i}(\boldsymbol{o},\boldsymbol{a})$ can also be replaced by the advantage function $A_{\zeta^i}(\boldsymbol{o},\boldsymbol{a}) := Q_{\zeta^i}(\boldsymbol{o},\boldsymbol{a}) - \mathbb{E}_{\boldsymbol{a}}[Q_{\zeta^i}(\boldsymbol{o},\boldsymbol{a})]$.

When evaluation in a novel MG, θ^i and ϕ^i is fine-tuned to greedily maximize agents' individual rewards. Denote $\omega^i = \{\theta^i, \phi^i\}$. Then the gradient of ω^i is given by

$$\nabla_{\omega^i}\mathcal{L}(\boldsymbol{\pi}) = \mathop{\mathbb{E}}_{\boldsymbol{o}\sim D, \boldsymbol{a}\sim\boldsymbol{\pi}}\left[\nabla_{\omega^i}\log\pi_{\theta^i}(a^i \mid o^i, \phi^i(o^i, z^i)) Q_{\zeta^i}(\boldsymbol{o},\boldsymbol{a})\right]. \tag{10}$$

In role-symmetric games, the parameters θ, ϕ, ζ are shared by homogeneous agents. And ϕ can be implemented as the option architecture (Sutton et al., 1999), i.e., g corresponds to the z-th option sampled from the categorical distribution. Implementation details and the variants can be found in Appendix E.

6.2 Mutual Information Maximization

In an iteration, several update steps of actor and critic are followed by mutual information $\mathcal{I}(g^i; a^i \mid o^i) + \mathcal{I}(m; g^i \mid o^i)$ maximization. According to the definition of mutual information and the non-negativeness of KL divergence, the following bound holds. And ϕ is optimized to optimize Equation 11 by gradient ascent. All the derivations in this section are provided in Appendix C.

$$\mathcal{I}(g^i; a^i \mid o^i) \geq \mathop{\mathbb{E}}_{g^i, o^i \sim D, a^i \sim \pi_{\theta^i}(\cdot \mid o^i, g^i)} \left[\log \frac{\pi_{\bar{\theta}^i}(a^i \mid o^i, g^i)}{p(a^i \mid o^i)} \right], \tag{11}$$

where $p(a^i \mid o^i) = \mathbb{E}_{z' \sim p(\cdot \mid m), g' = \phi^i(o^i, z')} \left[\pi_{\bar{\theta}}(a^i \mid o^i, g') \right]$.

The second mutual information term $\mathcal{I}(m; g^i \mid o^i)$ can be simplified by

$$\mathcal{I}(m; g^i \mid o^i) = \mathbb{E}_{m, o^i \sim D} \left[\log p(m \mid o^i, g^i) \right] + \log |\mathcal{M}|, \tag{12}$$

where $|\mathcal{M}|$ is the number of training MGs. To calculate the RHS of Equation 12, we introduce an auxiliary inference network ξ. Denote the one-hot index of MG as x. Then the auxiliary network outputs the index prediction $\hat{x}$, i.e., $p(\hat{x} \mid o, g; \xi)$. The cross-entropy objective is given as follows:

$$\min_{\psi, \xi} \mathbb{E}_{z \sim p(\cdot \mid m; \psi), o^i \sim D} \left[-x \log \left(p \left(\hat{x} \mid o^i, \phi^i(o^i, z); \xi \right) \right) \right]. \tag{13}$$

By minimizing Equation 13, ψ and ξ are simultaneously optimized.

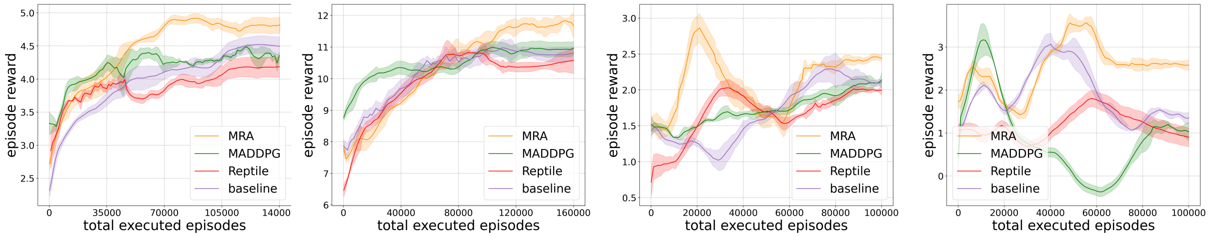
7 Experiments

Experiments are conducted in three environments built upon the particle-world framework (Lowe et al., 2017) that cover both competitive and mixed games, including the treasure collection environment, resource occupation environment, and the Pacman-like world. We provide the experiment settings and task descriptions to Appendix E.1.

7.1 Benefits of game-common Strategic Knowledge

We first conduct experiments to show the benefits of the proposed method by demonstrating that when training in various Markov Games, the game-common strategic knowledge can benefit individual training games.

In Figure 2, we compare MRA and the following baseline methods: (1) the MADDPG algorithm (Lowe et al., 2017) with relational representations (**MADDPG**); (2) the Reptile algorithm (Nichol et al., 2018) (**Reptile**); (3) baseline with the same network architectures as MRA, but agents learn their policies only in a *single* MG (**baseline**).



(a) Treasure collection.

(b) Resource occupation.

(c) Pacman-like world: **Left:** Pac-Man. **Right:** Ghost.

Figure 2: Benefits of meta-representations in the three environments. **(a):** 6 collector agents and 20 treasure dots; **(b):** 12 agents in the 6-resources environment; **(c):** 8 Pac-Man agents, 4 ghost agents and 20 food dots.

For MRA and Reptile, the size of training MG set $|\mathcal{M}|$ for the three environments is $4, 4, 3$, respectively. The settings of the training Markov Games in these three environments are as follows. For the treasure collection task, there are 4 training MGs. The numbers of agents are 3, 6, 12, and 24, respectively, which we denote as $\{3, 6, 12, 24\}$. The setting of the 4 training MGs in resource occupation task is $\{6, 9, 12, 15\}$. For the PacMan task, there are 3 training MGs in total: $\{(4, 2), (6, 3), (8, 4)\}$, where $(4, 2)$ in the first MG denotes that there are 4 PacMan agents and 2 ghost agents.

We note that the *actual* executed episodes of MRA in one MG are $|\mathcal{M}|$ times *smaller* than that for baseline and MADDPG, which reveals the efficiency of the proposed meta-representation. Comparisons between MRA and Reptile validate the effectiveness of MRA to meta-represent effective policies in various MGs. Due to the existence of game-specific knowledge, the optimal policies in different games are distinct. Thus, in Reptile, using a unimodal policy to represent them all negatively affect the performance. By comparing MRA with baseline and MADDPG, the benefit of the game-common strategic knowledge for individual games is revealed. Since the baseline agents are trained only in a single game, they cannot benefit from such common knowledge, even with more episodes executed. The common knowledge helps MRA outperform MADDPG. We also evaluate MADDPG with homogeneous agents sharing parameters, which works poorly. Although the Pacman-like world is not a zero-sum game, we still provide cross-comparison results in Appendix E.2 for completeness.

7.2 Performance Comparison in Multiple Games

In this part, we compare the performance of MRA with multi-task and meta-learning methods, including **EPC** (Long et al., 2020) and **RL^2** (Duan et al., 2016). Specifically, EPC learns policies from multiple training MGs with relational representations and evolutionary algorithms. However, EPC *refits* each training MG after obtaining effective policies in the previous training MG. On the contrary, MRA and RL^2 learn the generalizable strategic knowledge and essence-capturing prior knowledge, respectively.

The EPC algorithm, one of the curriculum learning approaches, is implemented by initializing 3 parallel sets of agents and mix-and-match the top 2 sets to the successive MG. For the meta-RL algorithm RL^2, each trial contains a cycle of all the $|\mathcal{M}|$ MGs. We also compare another MRA variant, **uni-MRA**, that samples z from a uniform distribution. In the treasure collection and resource occupation tasks, the results are shown in Figure 3.

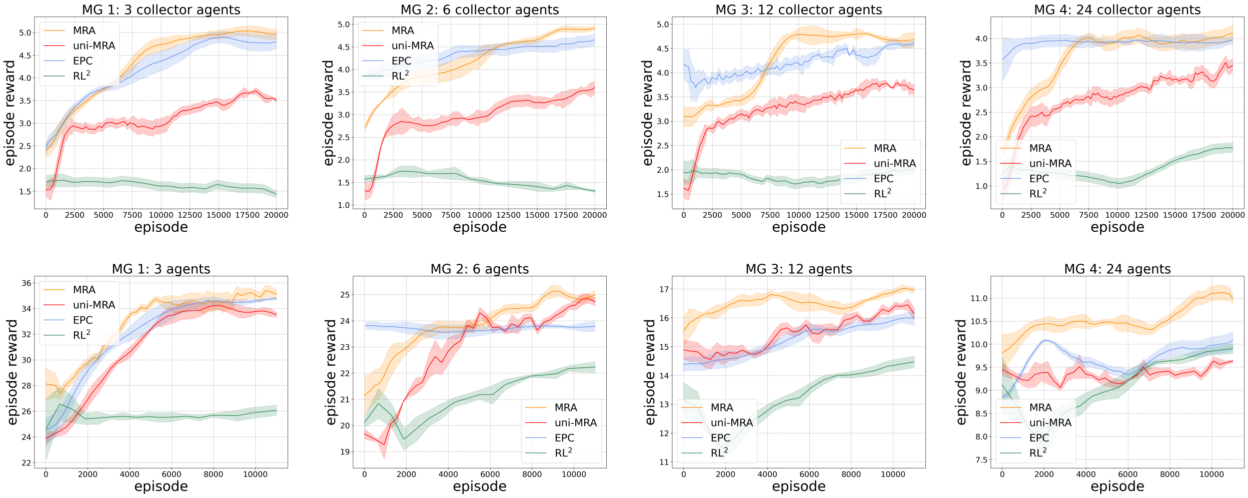


Figure 3: The training curves in the Markov Games constructed by varying the number of agents from the underlying environments. **First row:** Treasure collection environment; **Second row:** Resource occupation environment. In both environments, the total numbers of training MGs are 4, where the numbers of agents are 3, 6, 12, and 24, respectively.

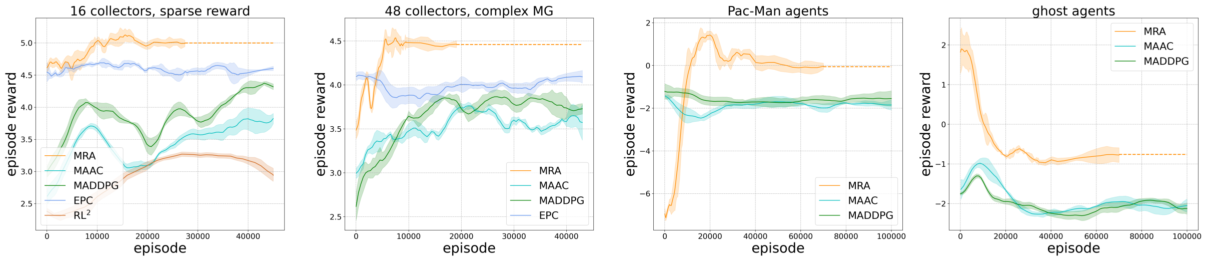
Due to the discrepancy between effective policies in different MGs, the game-common strategic knowledge is not well exploited by EPC. RL^2 agents are also observed to perform poorly in some MGs, which verifies the benefits of *explicitly* modeling the game-common knowledge and game-specific knowledge when the number of agents varies. Since no game-specific information is conditioned in uni-MRA, we observe that some MGs dominate others.

7.3 Generalization Evaluation

If agents with the learned policies can both (1) adapt better and faster; and (2) perform well in novel (or unseen) evaluation games in a zero-shot manner, then the algorithm is considered to generalize well. In the following parts, we test the generalizability of MRA and other baseline methods based on these two metrics.

Adaptation Ability: We first show that MRA adapts faster and better compared with **EPC**, **RL^2**, **MADDPG**, and **MAAC** (Iqbal & Sha, 2018). Here, MADDPG and MAAC are trained from scratch, while MRA, EPC, and RL^2 are fine-tuned from the parameters trained in multiple MGs as described in Section 7.2. The results are shown in Figure 4.

Our first observation is that the game-common knowledge can provide agents with a good policy initialization and thus overcome the random exploration bottleneck. For example, random exploration can happen in the Pacman-like world where the ghost agents populationally dominate the game. When this is the case, the episode will quickly terminate since the Pac-Man agents will soon be killed by the ghost agents. As a result of the lack of informative training signals, both the Pac-Man and ghost agents will end up with almost random behaviors. This is evidenced by the poor performance of the MAAC and MADDPG algorithms in Figure 4(c), where 8 ghost agents are chasing 2 Pac-Man



(a) Sparse rewards. (b) Large population. (c) Imbalanced competitive agents.

Figure 4: Adaptation performance comparison. **(a):** Sparse reward MG in the treasure collection environment; **(b):** Complex MG with a large number of agents in the treasure collection environment; **(c):** Imbalanced Pac-Man and ghost agents: 2 Pac-Man vs 8 ghosts, where the random exploration bottleneck prevents the agents to learn useful strategies due to the quickly terminated episodes and the lack of informative training signals. The dashed lines represent the asymptotic performance at convergence.

agents. In the contrast, with the extracted common knowledge guiding the Pac-Man to take ghost-avoidance actions, the MRA agents can easily learn to accomplish the task.

Besides, in Figure 4(a) when reward shaping is removed, i.e., in the sparse reward setting, MRA agents are able to effectively adapt to policies that have higher asymptotic performance in fewer episodes, compared to other baselines. We also show in Figure 4(b) that the complexity brought by the large population, such as 48, can be successfully handled by our MRA algorithm. These results verify the generalization advantage of the meta-representation in MRA compared with EPC, which only transfers knowledge in the *training* Markov Games.

Zero-Shot Transfer: MRA also has better generalizability compared to EPC and RL^2. In Figure 5, we report the ability of MRA to represent different policies in multiple training games and zero-shot transfer to evaluation games in the resource occupation environment. The performance of MRA is calculated by taking expectations over the latent z. We note that the return of MRA will become higher if enumerated trial-and-error is performed, i.e., choose the best policy mode among various latent samples z. Besides, we also provide additional experimental results and ablation studies in Appendix E.

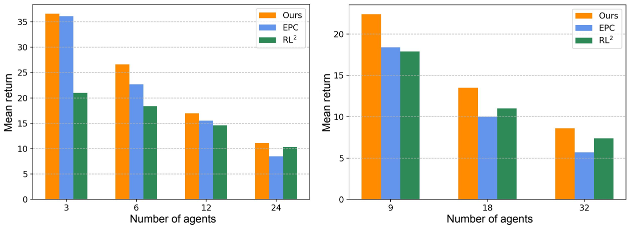


Figure 5: **Left:** Evaluation in the training MGs. **Right:** Zero-shot transfer to unseen evaluation MGs.

8 Conclusion & Discussions

In this paper, we propose meta representations for agents (MRA) that can generalize in Markov Games with varying populations. With latent variable policies and relational representations, the diverse strategic modes are captured. As an approximation to a theoretically justified objective, MRA effectively discovers the underlying strategic structures in the games that facilitate generalizable knowledge learning. Experimental results also verify the benefits of MRA.

Our work also opens several new problems. Theorem 1 requires the computation of ς and ϵ as well as an extremely large latent space, both of which are impractical. Although approximations that MRA makes are reasonable, obtaining optimal $\mathbf{\Pi}^*$ will not always be guaranteed. Possible future investigations include bounding ς by imposing restrictions on the evaluation MG set, or enlarging the latent space size by e.g., adopting continuous latent variables.

With role-symmetric game settings, MRA has benefits in many research problems, including dealing with population complexity, overcoming the multi-agent random exploration bottleneck, and adapting faster with the meta-represented agents. A fruitful avenue for future work is to augment MRA by e.g., adapting roles (Wang et al., 2020a), to apply to other game settings. Besides, achieving NE may not indicate the global optimality in general-sum MGs. Therefore, we would like to explore different metrics such as social optimum and correlated equilibrium as future work, which may require introducing the definition of distances established for these solution concepts.

For population-varying MGs, we model game-specific knowledge as strategic relationship. Although it may lose the universality in broader scopes compared with general meta-RL algorithms, we hope the idea of explicit strategic knowledge modeling can inspire algorithms that adjust with the task.

References

Akshat Agarwal, Sumit Kumar, and Katia Sycara. Learning transferable cooperative behavior in multi-agent teams. *arXiv preprint arXiv:1906.01202*, 2019.

Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. *arXiv preprint arXiv:1309.6821*, 2013.

Christos Dimitrakakis, David Parkes, Goran Radanovic, and Paul Tylkin. Multi-view decision processes: the helper-ai problem. In *31st Conference on Neural Information Processing Systems (NIPS)*. Neural Information Processing Systems Foundation, 2019.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*, 2017.

Jakob N Foerster, Yannis M Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. *arXiv preprint arXiv:1810.02912*, 2018.

Shariq Iqbal, Christian A Schroeder De Witt, Bei Peng, Wendelin Böhmer, Shimon Whiteson, and Fei Sha. Randomized entity-wise factorization for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4596–4606. PMLR, 2021.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in neural information processing systems*, pp. 7254–7264, 2018.

Michael Johanson, Kevin Waugh, Michael Bowling, and Martin Zinkevich. Accelerating best response calculation in large extensive games. In *IJCAI*, volume 11, pp. 258–265, 2011.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33, 2020.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4190–4203, 2017.

Andrzej Lasota and Michael C Mackey. *Chaos, fractals, and noise: stochastic aspects of dynamics*, volume 97. Springer Science & Business Media, 1998.

Bryan Lim, Luca Grillotti, Lorenzo Bernasconi, and Antoine Cully. Dynamics-aware quality-diversity for efficient learning of skill repertoires. *arXiv preprint arXiv:2109.08522*, 2021.

Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Animashree Anandkumar. Coach-player multi-agent reinforcement learning for dynamic team composition. *arXiv preprint arXiv:2105.08692*, 2021a.

Boyi Liu, Zhuoran Yang, and Zhaoran Wang. Policy optimization in zero-sum markov games: Fictitious self-play provably attains nash equilibria, 2021b. URL https://openreview.net/forum?id=c3MWGN_cTf.

Qian Long, Zihan Zhou, Abhinav Gupta, Fei Fang, Yi Wu†, and Xiaolong Wang†. Evolutionary population curriculum for scaling multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJxbHkrKDH.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pp. 6379–6390, 2017.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, pp. 7204–7213. PMLR, 2021.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, pp. 7611–7622, 2019.

Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, Zhe Wang, Guy Lever, Nicolas Heess, Thore Graepel, and Remi Munos. A generalized training approach for multiagent learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Bkl5kxrKDr.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*, pp. 2681–2690. PMLR, 2017.

Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

Julien Pérolat, Florian Strub, Bilal Piot, and Olivier Pietquin. Learning nash equilibrium for general-sum markov games from batch data. In *Artificial Intelligence and Statistics*, pp. 232–241. PMLR, 2017.

Goran Radanovic, Rati Devidze, David C Parkes, and Adish Singla. Learning to collaborate in markov decision processes. *arXiv preprint arXiv:1901.08029*, 2019.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised skill discovery. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. Neural mmo: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784*, 2019.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *Advances in neural information processing systems*, pp. 2244–2252, 2016.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Du, Yu Wang, and Yi Wu. Discovering diverse multi-agent strategic behavior via reward randomization. *arXiv preprint arXiv:2103.04564*, 2021.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4496–4506, 2017.

Justin K Terry, Nathaniel Grammel, Ananth Hari, Luis Santos, Benjamin Black, and Dinesh Manocha. Parameter sharing is surprisingly useful for multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18 (2):77–95, 2002.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*, 2019.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020a.

Weixun Wang, Tianpei Yang, Yong Liu, Jianye Hao, Xiaotian Hao, Yujing Hu, Yingfeng Chen, Changjie Fan, and Yang Gao. From few to more: Large-scale dynamic multiagent curriculum learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7293–7300, 2020b.

Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst continual structured non-stationarity. In *International Conference on Machine Learning*, pp. 11393–11403. PMLR, 2021.

Huanhuan Yang, Dianxi Shi, Chenran Zhao, Guojun Xie, and Shaowu Yang. Ciexplore: Curiosity and influence-based exploration in multi-agent cooperative scenarios with sparse rewards. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2321–2330, 2021.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. *arXiv preprint arXiv:1802.05438*, 2018.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Policy optimization provably converges to nash equilibria in zero-sum linear quadratic games. *arXiv preprint arXiv:1906.00729*, 2019.

Shenao Zhang. Conservative dual policy optimization for efficient model-based reinforcement learning. *arXiv preprint arXiv:2209.07676*, 2022.

Shenao Zhang, Li Shen, Zhifeng Li, and Wei Liu. Structure-regularized attention for deformable object representation. *arXiv preprint arXiv:2106.06672*, 2021.

Stephan Zheng and Yisong Yue. Structured exploration via hierarchical variational policy networks, 2018. URL https://openreview.net/forum?id=HyunpgbR-.

A Proofs

A.1 Proof Sketch of Theorem 1

Theorem 1 can be proved by establishing the following lemmas.

Lemma 1. *Define the distance $\kappa(\pi^i)$ between policy π^i and the best response π^{*i} in the action space as:*

$$\kappa(\pi^i) = \left\| \left\| \pi^{*i}(a \mid s) - \pi^i(a \mid s) \right\|_{a,1} \right\|_{s,\infty}.$$

For any joint strategy $\boldsymbol{\pi}$, the NashConv *$\mathcal{D}_m(\boldsymbol{\pi})$ is bounded by:*

$$\mathcal{D}_m(\boldsymbol{\pi}) \le \left(\frac{\gamma \iota_m}{(1-\gamma)^2} + \frac{1}{1-\gamma} \right) \left\| \kappa(\pi^i) \right\|_{i,1}.$$

Lemma 1 builds a bridge between the action-space distance and the value-space distance $\mathcal{D}_m(\boldsymbol{\pi})$. Then the distance ς between $\mathcal{M}$ and $\mathcal{M}'$ can be represented in form of $\kappa(\pi^i)$ for some specific index i. Intuitively, if ϵ is larger than a certain threshold, the ϵ-range joint strategy set $\hat{\boldsymbol{\Pi}}$ is also large enough to contain all the strategies that achieve NE in evaluation games. Formally, we provide Lemma 2.

Lemma 2. *For the training MG set $\mathcal{M}$ and the evaluation MG set $\mathcal{M}'$, if ϵ satisfies*

$$\epsilon \ge \varsigma - \min_{\iota_m, \iota_{m'}} \frac{\varsigma \gamma \left(\iota_{m'} - \iota_m \right)}{\gamma \iota_{m'} + 1 - \gamma}, \tag{14}$$

then for every evaluation Markov Game $m' \in \mathcal{M}'$, the joint strategy that achieves Nash Equilibrium in m' is guaranteed to be contained in the ϵ-range joint strategy set $\hat{\boldsymbol{\Pi}}$.

We then show how Lemma 2 can be utilized to obtain the objective in Equation 7. The first step is to find an equivalence with optimization objective as formally stated in Lemma 3.

Lemma 3. *If $|\boldsymbol{\Pi}_\Theta| \ge |\hat{\boldsymbol{\Pi}}|$ and ϵ satisfies $\epsilon \ge \varsigma - \min_{\iota_m, \iota_{m'}} \frac{\varsigma\gamma(\iota_{m'} - \iota_m)}{\gamma \iota_{m'} + 1 - \gamma}$, then the optimal $\boldsymbol{\Pi}$ that maximizes the objective:*

$$\mathcal{L}(\boldsymbol{\Pi}) = \max_{\boldsymbol{\pi} \sim \boldsymbol{\Pi}} \min_{\hat{\boldsymbol{\pi}} \sim \hat{\boldsymbol{\Pi}}} \mathbb{E}_{\boldsymbol{a} \sim \hat{\boldsymbol{\pi}}, \boldsymbol{o}} \left[\log \boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{o}) \right], \tag{15}$$

for every evaluation Markov Game $m' \in \mathcal{M}'$, there exists a joint strategy $\boldsymbol{\pi} \in \boldsymbol{\Pi}_{\Theta^*}$ that reaches Nash Equilibrium (i.e., $\boldsymbol{\Pi}_{\Theta^*} = \boldsymbol{\Pi}^*$ satisfies equation Equation 2).

The next step is to build the relationship between Equation 15 and the mutual information objective in Equation 7. If $\boldsymbol{\pi} \in \hat{\boldsymbol{\Pi}}$, we notice that $\max_{\boldsymbol{\pi}} \mathbb{E}_{\hat{\boldsymbol{\pi}}} [\log \boldsymbol{\pi}] = -\mathcal{H}(\hat{\boldsymbol{\pi}})$. Thus, Theorem 1 can be proved by leveraging the close connection between entropy and mutual information.

In the sequel, we provide the proofs of the above three lemmas in A.2, A.3, and A.4, respectively. The proof of Theorem 1 in A.5 is built upon these lemmas.

A.2 Proof of Lemma 1

Before giving the proof of Lemma 1, we first provide a useful lemma of the Markov state transition operator $\mathcal{P}_m^{\pi^i, \boldsymbol{\pi}^{\text{-}i}}$. The following Lemma 4 is modified from (Liu et al., 2021b) Lemma E.1, which is originally presented for fictitious self-play. We modify the lemma and the proof to be suitable for the concerned multi-agent general-sum Markov Game setting with our notations.

Lemma 4. *(Liu et al., 2021). The Markov state transition operator satisfies:*

$$\left\| \sum_t \gamma^t \left[\left(\mathcal{P}_m^{\pi^{*i}, \boldsymbol{\pi}^{\text{-}i}} \right)^t - \left(\mathcal{P}_m^{\pi^i, \boldsymbol{\pi}^{\text{-}i}} \right)^t \right] \right\|_{op} \le \frac{\gamma}{(1-\gamma)^2} \left\| \mathcal{P}_m^{\pi^{*i}, \boldsymbol{\pi}^{\text{-}i}} - \mathcal{P}_m^{\pi^i, \boldsymbol{\pi}^{\text{-}i}} \right\|_{op}.$$

Proof. As a first step, we study the operator $\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}$ and obtain the following results:

$$\begin{aligned}
&\left\| \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t \right\|_{op} \\
&= \left\| \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1} \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right) + \left(\left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1} - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1}\right) \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op} \\
&\le \left\| \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1} \left(\mathcal{P}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right) \right\|_{op} + \left\| \left(\left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1} - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1}\right) \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op} \\
&\le \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op} + \left\| \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1} - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-1} \right\|_{op},
\end{aligned} \tag{16}$$

where the equality follows from basic algebra, the first inequality follows from the triangle inequality, and the last inequality holds since $\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} \le 1$ (Lasota & Mackey, 1998).

Recursively applying Equation 16, we obtain

$$\begin{aligned}
&\left\| \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t \right\|_{op} \\
&\le \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op} + \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op} + \left\| \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-2} - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{t-2} \right\|_{op} \\
&\le t \cdot \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op}.
\end{aligned}$$

Then we have by summation that

$$\begin{aligned}
\left\| \sum_t \gamma^t \left[\left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t \right] \right\|_{op} &\le \sum_t \gamma^t \left\| \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t - \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t \right\|_{op} \\
&\le \left(\sum_t t\gamma^t\right) \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op} \\
&= \frac{\gamma}{(1-\gamma)^2} \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{op}.
\end{aligned}$$

This concludes the proof of Lemma 4. □

Now we are ready to prove Lemma 1.

Proof. To begin, we denote by $\rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}$ the state visitation measure of the joint strategy $(\pi^{*i}, \boldsymbol{\pi}^{-\boldsymbol{i}})$ in the Markov Game m, which is defined as follows:

$$\begin{aligned}
\rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} &= \left(\mathcal{I} - \gamma \mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^{-1} \delta_s \\
&= \left(\sum_t \gamma^t \left(\mathcal{P}_m^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}\right)^t\right) \delta_s,
\end{aligned}$$

where δ_s is a Dirac delta function.

By converting the value of strategy to the integration of reward over state measure, it holds that

$$\begin{aligned}
&v_{\boldsymbol{\pi}^{-\boldsymbol{i}}}^{*i,m}(s) - v_{\boldsymbol{\pi}}^{i,m}(s) \\
&= \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a})\right] - \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a})\right] \\
&= \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a})\right] - \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a})\right] \\
&\qquad + \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a}) - \mathbb{E}_{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a})\right] \\
&\le \left\| \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} - \rho_{s,m}^{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} \right\|_{s,1} + \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^{*i},\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a}) - \mathbb{E}_{\pi^i,\boldsymbol{\pi}^{-\boldsymbol{i}}} r_m^i(s', \boldsymbol{a})\right],
\end{aligned} \tag{17}$$

where the inequality follows from the definition of the $\mathcal{L}_1$-norm over the state space.

We then bound the resulting two terms separately. For the first term, we have from Lemma 4 that:

$$\left\| \sum_t \gamma^t \left[\left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t - \left(\mathcal{P}_m^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t \right] \right\|_{op} \leq \frac{\gamma}{(1-\gamma)^2} \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} - \mathcal{P}_m^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right\|_{op}$$

Following from the definition of ι_m, we have

$$\begin{aligned} \left\| \sum_t \gamma^t \left[\left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t - \left(\mathcal{P}_m^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t \right] \right\|_{op} &\leq \frac{\gamma}{(1-\gamma)^2} \left\| \mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} - \mathcal{P}_m^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right\|_{op} \\ &\leq \frac{\gamma \iota_m}{(1-\gamma)^2} \kappa(\pi^i) \end{aligned}$$

Thus, it holds that

$$\begin{aligned} \left\| \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} - \rho_{s,m}^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right\|_{s,1} &= \left\| \left(\sum_t \gamma^t \left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t \right) \delta_s - \left(\sum_t \gamma^t \left(\mathcal{P}_m^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t \right) \delta_s \right\|_{s,1} \\ &\leq \left\| \sum_t \gamma^t \left[\left(\mathcal{P}_m^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t - \left(\mathcal{P}_m^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right)^t \right] \right\|_{op} \cdot \|\delta_s\|_{s,1} \\ &\leq \frac{\gamma \iota_m}{(1-\gamma)^2} \kappa(\pi^i), \end{aligned} \tag{18}$$

where the last inequality holds due to Assumption 1.

Besides, the second term on the right-hand side of Equation 17 satisfies

$$\begin{aligned} &\left\| \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} r_m^i(s', \boldsymbol{a}) - \mathbb{E}_{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} r_m^i(s', \boldsymbol{a}) \right] \right\|_{s,\infty} \\ &\leq \left\| \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}}} \left[\left\| \pi^{*i}(\cdot \mid s) - \pi^i(\cdot \mid s) \right\|_1 \right] \right\|_{s,\infty} \\ &\leq \kappa(\pi^i) \left(\sum_t \gamma^t \right) = \frac{\kappa(\pi^i)}{1-\gamma}, \end{aligned} \tag{19}$$

where the second inequality follows from the definition of $\kappa(\pi^i)$.

Finally, combining Equation 18 and Equation 19, we obtain the stated result for $\mathcal{D}_m(\boldsymbol{\pi})$ as follows:

$$\begin{aligned} \mathcal{D}_m(\boldsymbol{\pi}) &= \left\| \left\| v_{\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}}^{*i,m}(s) - v_{\boldsymbol{\pi}}^{i,m}(s) \right\|_{s,\infty} \right\|_{i,1} \\ &\leq \left\| \left\| \left\| \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} - \rho_{s,m}^{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} \right\|_{s,1} + \mathbb{E}_{s' \sim \rho_{s,m}^{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}}} \left[\mathbb{E}_{\pi^{*i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} r_m^i(s', \boldsymbol{a}) - \mathbb{E}_{\pi^{i},\boldsymbol{\pi}^{\text{-}\boldsymbol{i}}} r_m^i(s', \boldsymbol{a}) \right] \right\|_{s,\infty} \right\|_{i,1} \\ &\leq \left\| \frac{\gamma \iota_m}{(1-\gamma)^2} \kappa(\pi^i) + \frac{\kappa(\pi^i)}{1-\gamma} \right\|_{i,1} \\ &= \left(\frac{\gamma \iota_m}{(1-\gamma)^2} + \frac{1}{1-\gamma} \right) \left\| \kappa(\pi^i) \right\|_{i,1} \end{aligned}$$

□

A.3 Proof of Lemma 2

Proof. Definition 1 describes the distance ς between the training MG set $\mathcal{M}$ and the evaluation MG set $\mathcal{M}'$. In the following, we provide an equivalent logic statement:

$$\begin{gathered}\forall m' \in \mathcal{M}', i \in \{1, \ldots, N_{m'}\}, \exists m \in \mathcal{M}, \boldsymbol{\pi}, \boldsymbol{\pi'}, i' \in h_{i,m}, \\ \text{s.t. } \mathcal{D}_m(\boldsymbol{\pi}) = 0, \mathcal{D}_{m'}(\boldsymbol{\pi'}) = 0, \forall \pi^i \in \boldsymbol{\pi}, \mathcal{D}_{m'}(\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}}) \leq \varsigma.\end{gathered}$$

In an evaluation MG $\tilde{m}' \in \mathcal{M}'$, let i and i' be the agent index defined as follows:

$$i = \underset{i \in \{1, \ldots, N_{\tilde{m}'}\}}{\arg\max} \left[\min_{i' \in h_{i,m}} \mathcal{D}_{\tilde{m}'}(\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}}) \right], \text{ s.t. } \mathcal{D}_m(\boldsymbol{\pi}) = 0, \mathcal{D}_{\tilde{m}'}(\boldsymbol{\pi'}) = 0 \tag{20}$$

Intuitively, the above agent i is the agent in $\tilde{m}'$ that being replaced by the trained policy $\pi^{i'}$ in the policy set leads to the largest distance to the Nash Equilibrium. And agent i' is the corresponding agent that $\pi^{i'}$ achieves an NE in a particular training MG.

With i and i' denied in Equation 20, the bound in Lemma 1 can be specified as follows:

$$\begin{aligned}\mathcal{D}_{\tilde{m}'}(\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}}) &= \left\| \left\| v^{*i,\tilde{m}'}_{\boldsymbol{\pi'^{\text{-}i}}} - v^{i,\tilde{m}'}_{\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}}} \right\|_{s,\infty} \right\|_{i,1} \\ &= \left\| v^{i,\tilde{m}'}_{\boldsymbol{\pi'}} - v^{i,\tilde{m}'}_{\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}}} \right\|_{s,\infty} \\ &= \left\| v^{i,\tilde{m}'}_{\pi'^i, \boldsymbol{\pi'^{\text{-}i}}} - v^{i,\tilde{m}'}_{\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}}} \right\|_{s,\infty} \\ &\leq \frac{\gamma \iota_{\tilde{m}'}}{(1-\gamma)^2} \kappa(\pi^{i'}) + \frac{\kappa(\pi^{i'})}{1-\gamma},\end{aligned}$$

where the second equality holds since $\mathcal{D}_{\tilde{m}'}(\boldsymbol{\pi'}) = 0$, and the distance from the joint strategy $(\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}})$ to a Nash Equilibrium is equal to the distance to $\boldsymbol{\pi'}$. The last inequality holds due to the bound in Lemma 1 and the definition of i and i'.

This implies that for any MG $\tilde{m}'$, it holds that

$$\max_{i \in \{1, \ldots, N_{\tilde{m}'}\}} \min_{\substack{m \in \mathcal{M}, i' \in h_i \\ \boldsymbol{\pi} \in \{\boldsymbol{\pi} | \mathcal{D}_m(\boldsymbol{\pi}) = 0\} \\ \boldsymbol{\pi'} \in \{\boldsymbol{\pi'} | \mathcal{D}_{\tilde{m}'}(\boldsymbol{\pi'}) = 0\}}} \mathcal{D}_{\tilde{m}'}(\pi^{i'}, \boldsymbol{\pi'^{\text{-}i}}) \leq \frac{\gamma \iota_{\tilde{m}'}}{(1-\gamma)^2} \kappa(\pi^{i'}) + \frac{\kappa(\pi^{i'})}{1-\gamma}. \tag{21}$$

With the maximum influence $\iota_{m'}$ over the evaluation MG $m' \in \mathcal{M}'$, we obtain:

$$\varsigma = \max_{\iota_{m'}} \frac{\gamma \iota_{m'}}{(1-\gamma)^2} \kappa(\pi^{i'}) + \frac{\kappa(\pi^{i'})}{1-\gamma}. \tag{22}$$

Since $\mathcal{D}_m(\boldsymbol{\pi}) = 0$ and $\mathcal{D}_{m'}(\boldsymbol{\pi'}) = 0$, the best response in the Markov Game m with other agents' policies fixed as $\boldsymbol{\pi^{\text{-}i'}}$ is $\pi^{i'}$. Similarly, in game m', the best response with other agent's policies fixed as $\boldsymbol{\pi'^{\text{-}i}}$ is π'^i. Therefore, we obtain

$$\kappa(\pi^{i'}) = \kappa(\pi'^i) = \left\| \left\| \pi^{i'}(a) - \pi'^i(a) \right\|_{a,1} \right\|_{s,\infty}.$$

For ϵ that satisfies Equation 14, we obtain:

$$\begin{aligned}\epsilon &\geq \max_{\iota_{m'}, \iota_m} \varsigma - \frac{\varsigma \gamma \left(\iota_{m'} - \iota_m\right)}{\gamma \iota_{m'} + 1 - \gamma} \\ &= \max_{\iota_{m'}, \iota_m} \left(\frac{\gamma \iota_m + 1 - \gamma}{\gamma \iota_{m'} + 1 - \gamma} \right) \cdot \left(\frac{\gamma \iota_{m'}}{(1-\gamma)^2} \kappa(\pi^{i'}) + \frac{\kappa(\pi^{i'})}{1-\gamma} \right) \\ &\geq \max_{\iota_m} \left(\frac{\gamma \iota_m + 1 - \gamma}{\gamma \iota_{\tilde{m}'} + 1 - \gamma} \right) \cdot \left(\frac{\gamma \iota_{\tilde{m}'}}{(1-\gamma)^2} \kappa(\pi'^i) + \frac{\kappa(\pi'^i)}{1-\gamma} \right) \\ &\geq \max_{\iota_m} \frac{\gamma \iota_m}{(1-\gamma)^2} \kappa(\pi'^i) + \frac{\kappa(\pi'^i)}{1-\gamma},\end{aligned} \tag{23}$$

where the equality follows from Equation 22, and the last two inequalities follow from basic algebra.

Thus, we obtain

$$\begin{aligned}
\mathcal{D}_m\left((\pi^{i'}, \boldsymbol{\pi}^{\text{-}\boldsymbol{i}'})\right) &= \left\|\left\|v^{*i,m}_{\boldsymbol{\pi}^{\text{-}\boldsymbol{i}'}} - v^{i,m}_{\pi'^{i}, \boldsymbol{\pi}^{\text{-}\boldsymbol{i}'}}\right\|_{s,\infty}\right\|_{i,1} \\
&= \left\|v^{i,m}_{\boldsymbol{\pi}} - v^{i,m}_{\pi'^{i}, \boldsymbol{\pi}^{\text{-}\boldsymbol{i}'}}\right\|_{s,\infty} \\
&= \left\|v^{i,m}_{\pi^{i'}, \boldsymbol{\pi}^{\text{-}\boldsymbol{i}'}} - v^{i,m}_{\pi'^{i}, \boldsymbol{\pi}^{\text{-}\boldsymbol{i}'}}\right\|_{s,\infty} \\
&\leq \frac{\gamma \iota_m}{(1-\gamma)^2}\kappa(\pi'^{i}) + \frac{\kappa(\pi'^{i})}{1-\gamma} \\
&\leq \max_{\iota_m} \frac{\gamma \iota_m}{(1-\gamma)^2}\kappa(\pi'^{i}) + \frac{\kappa(\pi'^{i})}{1-\gamma} \\
&\leq \epsilon,
\end{aligned}$$

where the second equality holds by the definition of i and i' in Equation 20, the third equality holds since $\mathcal{D}_m(\boldsymbol{\pi}) = 0$, and the last inequality follows from Equation 23.

This indicates that if we choose ϵ satisfying Equation 14, then the policy π'^{i} that achieves Nash Equilibrium in the Markov Game $\tilde{m}'$ is guaranteed to be contained in the policy set.

Since all the above inequalities hold for any $\tilde{m}' \in \mathcal{M}'$, the policy that achieves NE in all MGs in the evaluation MG set $\mathcal{M}'$ is guaranteed to be included in the policy set. This completes the proof of Lemma 2. □

A.4 Proof of Lemma 3

Proof. We first know that

$$\mathbb{E}_{\boldsymbol{a}\sim\hat{\boldsymbol{\pi}},\boldsymbol{o}}\left[\log \hat{\boldsymbol{\pi}}(\boldsymbol{a} \mid \boldsymbol{o}) - \log \boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{o})\right] \geq 0.$$

Since $|\boldsymbol{\Pi}| \geq |\hat{\boldsymbol{\Pi}}|$, the optimal $|\boldsymbol{\Pi}|$ that maximizes $\mathcal{L}(\boldsymbol{\Pi})$ in Equation 15 must satisfy

$$\min_{\hat{\boldsymbol{\pi}}\sim\hat{\boldsymbol{\Pi}}} \mathbb{E}_{\boldsymbol{a}\sim\hat{\boldsymbol{\pi}},\boldsymbol{o}}\left[\log \boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{o})\right] \leq \mathbb{E}_{\boldsymbol{a}\sim\hat{\boldsymbol{\pi}},\boldsymbol{o}}\left[\log \hat{\boldsymbol{\pi}}(\boldsymbol{a} \mid \boldsymbol{o})\right] \leq 0.$$

In other words, for every policy $\hat{\boldsymbol{\pi}}$ in the joint strategy set $\hat{\boldsymbol{\Pi}}$, i.e., $\hat{\boldsymbol{\pi}} \in \hat{\boldsymbol{\Pi}}$, there exists a learned policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, such that $\boldsymbol{\pi} = \hat{\boldsymbol{\pi}}$. Note that the above statement is true only when $|\boldsymbol{\Pi}| \geq |\hat{\boldsymbol{\Pi}}|$.

For ϵ that satisfies Equation 14, from Lemma 2 we know that for every evaluation MG $m' \in \mathcal{M}'$, the strategy that achieves Nash Equilibrium are guaranteed to be contained in $\hat{\boldsymbol{\Pi}}$. Thus, the optimal policy set $\boldsymbol{\Pi}$ that results from optimizing Equation 15 also contains the strategies that are NE in every MG $m' \in \mathcal{M}'$.

So the optimal policy set $\boldsymbol{\Pi}$ satisfies Equation 2, which completes the proof.

□

A.5 Proof of Theorem 1

Proof. Since $|\boldsymbol{\Pi}_\Theta| \geq |\hat{\boldsymbol{\Pi}}|$, we can simplify the objective in Equation 15 by updating the joint strategy $\boldsymbol{\pi}$ in a fixed-size set $\boldsymbol{\Pi}$. That is, maximizing Equation 15 is equivalent to

$$\max_{\boldsymbol{\Pi}} \min_{\hat{\boldsymbol{\pi}}\sim\hat{\boldsymbol{\Pi}}} \max_{\boldsymbol{\pi}\sim\boldsymbol{\Pi}} \mathbb{E}_{\boldsymbol{a}\sim\hat{\boldsymbol{\pi}},\boldsymbol{o}}\left[\log \boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{o})\right] = \max_{\boldsymbol{\pi}\sim\boldsymbol{\Pi}} \min_{\hat{\boldsymbol{\pi}}\sim\hat{\boldsymbol{\Pi}}} \mathbb{E}_{\boldsymbol{a}\sim\hat{\boldsymbol{\pi}},\boldsymbol{o}}\left[\log \boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{o})\right]. \tag{24}$$

From the non-negativeness of KL divergence, we have:

$$\mathcal{D}_{KL}\left(\hat{\boldsymbol{\pi}}, \boldsymbol{\pi}\right) = \mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\log \frac{\hat{\boldsymbol{\pi}}}{\boldsymbol{\pi}}\right] \geq 0,$$

where the equality holds when $\boldsymbol{\pi} = \hat{\boldsymbol{\pi}}$.

Thus, we have from the definition of entropy that

$$\max_{\boldsymbol{\pi}} \mathbb{E}_{\hat{\boldsymbol{\pi}}} [\log \boldsymbol{\pi}] \leq -\mathcal{H}(\hat{\boldsymbol{\pi}}).$$

If $\boldsymbol{\pi} \in \hat{\boldsymbol{\Pi}}$ is constrained, then the equality holds and $\max_{\boldsymbol{\pi}} \mathbb{E}_{\hat{\boldsymbol{\pi}}} [\log \boldsymbol{\pi}] = -\mathcal{H}(\hat{\boldsymbol{\pi}})$. This leads to

$$\begin{aligned}\max_{\boldsymbol{\pi}\sim\boldsymbol{\Pi}} \min_{\hat{\boldsymbol{\pi}}\sim\hat{\boldsymbol{\Pi}}} \mathbb{E}_{\hat{\boldsymbol{\pi}}} [\log \boldsymbol{\pi}] &= \min_{\hat{\boldsymbol{\pi}}\sim\hat{\boldsymbol{\Pi}}} -\mathcal{H}(\hat{\boldsymbol{\pi}}) \\ &= \max_{\hat{\boldsymbol{\pi}}\sim\hat{\boldsymbol{\Pi}}} \mathcal{H}(\hat{\boldsymbol{\pi}}) \\ &= \max_{\boldsymbol{\pi}\sim\boldsymbol{\Pi}} \mathcal{H}(\boldsymbol{\pi}), \text{ s.t. } \boldsymbol{\pi} \in \hat{\boldsymbol{\Pi}}.\end{aligned} \tag{25}$$

The above equation states that the strategy $\boldsymbol{\pi}$ is learned to fit $\hat{\boldsymbol{\pi}}$. And to cover all the $\hat{\boldsymbol{\pi}} \in \hat{\boldsymbol{\Pi}}$, the entropy of strategies in $\boldsymbol{\Pi}$ should also be maximized.

Then we get the following equivalence:

$$\begin{aligned}\max_{\boldsymbol{\pi}\sim\boldsymbol{\Pi}} \mathcal{H}(\boldsymbol{\pi}) &= \max_{\Omega} \mathcal{I}(m; a \mid o) + \mathcal{H}(a \mid m, o) \\ &= \max_{\Omega} \mathcal{I}(m; a \mid o) + \mathcal{I}(g; a \mid m, o) + \mathcal{H}(a \mid m, o, g) \\ &= \max_{\Omega} \mathcal{I}(m; a \mid o) + \mathcal{I}(g; a \mid o) + \mathcal{H}(a \mid m, o, g) \\ &= \max_{\Omega} \mathcal{I}(m; a \mid o) + \mathcal{I}(g; a \mid o),\end{aligned} \tag{26}$$

where the first equality holds following the definition of mutual information and by noticing that policy $\boldsymbol{\pi}$ is meta-represented by Ω. The last equality holds since the size of the meta-represented policy set $\mid \boldsymbol{\Pi}_\Theta \mid$ is sufficiently large and satisfying $\mid \boldsymbol{\Pi}_\Theta \mid \geq \mid \hat{\boldsymbol{\Pi}} \mid$.

Combining Equation 25 and Equation 26, we have

$$\min_{\hat{\boldsymbol{\pi}}\sim\hat{\boldsymbol{\Pi}}} \max_{\boldsymbol{\pi}\sim\boldsymbol{\Pi}} \mathbb{E}_{\hat{\boldsymbol{\pi}}} [\log \boldsymbol{\pi}] = \max_{\Omega} \mathcal{I}(m; a \mid o) + \mathcal{I}(g; a \mid o), \text{ s.t. } \boldsymbol{\pi} \in \hat{\boldsymbol{\Pi}}. \tag{27}$$

Then by Equation 24 and Equation 27, we have that the solution of the following objective is equivalent to the solution of Equation 15, i.e.,

$$\psi^*, \phi^* = \arg\max_{\psi,\phi} \mathcal{I}(g; a \mid o) + \mathcal{I}(m; g \mid o) \text{ s.t. } \boldsymbol{\pi}_{\theta^*} \in \hat{\boldsymbol{\Pi}}.$$

Thus, for every evaluation Markov Game $m' \in \mathcal{M}'$, there exists a joint strategy $\boldsymbol{\pi} \in \boldsymbol{\Pi}_\Theta$ that reaches Nash Equilibrium (i.e., $\boldsymbol{\Pi}_{\Theta^*} = \boldsymbol{\Pi}^*$ satisfies Equation 2). □

B Fast Adaptation with First-Order Gradient

Reptile (Nichol et al., 2018) is a meta-learning algorithm that uses first-order gradient information for fast adaptation.

For parameter θ that maximizes objective $\mathcal{L}_m^k(\theta)$ in the k-th mini-batch of game m, θ is updated by $\theta \leftarrow \theta + \alpha\Delta\theta$, where $\Delta\theta = U_m^K(\theta) - \theta$ and $U_m^K(\theta)$ denotes the updated θ after K gradient steps with learning rate β, and α is a hyperparameter.

Denote the k-th step parameter as θ_k, then the update $\Delta\theta$ of K gradient steps is as follows:

$$
\begin{aligned}
\Delta\theta &= \theta_K - \theta_1 \\
&= \beta \sum_{k=1}^{K-1} \nabla \mathcal{L}_m^k(\theta_k) \\
&= \beta \sum_{k=1}^{K-1} \left(\nabla \mathcal{L}_m^k(\theta_1) + \nabla^2 \mathcal{L}_m^k(\theta_1)\left(\theta_k - \theta_1\right) + \mathcal{O}\left(\left\|\theta_k - \theta_1\right\|^2 \right) \right) \\
&= \beta \sum_{k=1}^{K-1} \left(\nabla \mathcal{L}_m^k(\theta_1) + \beta \nabla^2 \mathcal{L}_m^k(\theta_1) \sum_{j=1}^{k-1} \nabla \mathcal{L}_m^k(\theta_j) + \mathcal{O}\left(\beta^2\right) \right) \\
&= \beta \left[\sum_{k=1}^{K-1} \left(\nabla \mathcal{L}_m^k(\theta_1) + \beta \sum_{j=1}^{k-1} \left(\nabla^2 \mathcal{L}_m^k(\theta_1) \nabla \mathcal{L}_m^j(\theta_1) \right) \right) + \mathcal{O}\left(\beta^2\right) \right],
\end{aligned}
$$

where the last equation holds since $\nabla \mathcal{L}_m^k(\theta_j) = \nabla \mathcal{L}_m^k(\theta_1) + \mathcal{O}(\beta)$.

For the initial parameter $\theta = \theta_1$, the term $\sum_{k=1}^{K-1} \nabla \mathcal{L}_m^k(\theta_1)$ maximizes the overall performance at θ in all the K mini-batches in an MG m. The key difference from the joint training objective is the term $\nabla^2 \mathcal{L}_m^k(\theta_1) \nabla \mathcal{L}_m^j(\theta_1)$. When the expectation are taken under mini-batch sampling in m, we denote by $\mathbb{E}_k$ the expectation over the mini-batch defined by J^k. Omitting the higher-order term $\mathcal{O}\left(\beta^2\right)$, we have

$$
\begin{aligned}
\mathbb{E}\left[\Delta\theta\right] &= (K-1)\mathbb{E}_k\left[\nabla \mathcal{L}_m^k(\theta)\right] + (K-1)(K-2)\beta \cdot \mathbb{E}_{j,k}\left[\nabla^2 \mathcal{L}_m^k(\theta) \nabla \mathcal{L}_m^j(\theta)\right] \\
&= (K-1)\mathbb{E}_k\left[\nabla \mathcal{L}_m^k(\theta)\right] \\
&+ \frac{(K-1)(K-2)\beta}{2} \mathbb{E}_{j,k}\left[\nabla^2 \mathcal{L}_m^k(\theta) \nabla \mathcal{L}_m^j(\theta) + \nabla^2 \mathcal{L}_m^j(\theta) \nabla \mathcal{L}_m^k(\theta)\right] \\
&= (K-1)\mathbb{E}_k\left[\nabla \mathcal{L}_m^k(\theta)\right] + \frac{(K-1)(K-2)\beta}{2} \mathbb{E}_{j,k}\left[\nabla\left(\nabla \mathcal{L}_m^k(\theta) \nabla \mathcal{L}_m^j(\theta)\right)\right].
\end{aligned}
$$

Thus, updating θ by $\theta \leftarrow \theta + \alpha\Delta\theta$ not only maximizes the average performance in K mini-batches of all Markov Games, but also maximizes the inner product between gradients of different mini-batches, i.e., $\nabla \mathcal{L}_m^k(\theta) \nabla \mathcal{L}_m^j(\theta)$. Thus, the generalization ability is improved and fast adaptation is achieved.

C Derivation of Mutual Information Calculation

The two mutual information terms in Equation 7, i.e., $\mathcal{I}(g; a \mid o)$ and $\mathcal{I}(m; g \mid o)$ can be calculated as follows:

$$
\begin{aligned}
\mathcal{I}(g; a \mid o) &= \int p(a, o, g) \log \frac{p(a \mid o, g)}{p(a \mid o)} da\, do\, dg \\
&= \mathbb{E}_{a,o,g}[\log \frac{\pi_{\bar{\theta}}(a \mid o, g)}{p(a \mid o)}] + \mathbb{E}_{a,o,g}[\mathcal{D}_{KL}(p(a \mid o, g) \mid\mid \pi_{\bar{\theta}}(a \mid o, g))] \\
&\geq \mathbb{E}_{a,o,g}[\log \frac{\pi_{\bar{\theta}}(a \mid o, g)}{p(a \mid o)}] \\
&\approx \mathbb{E}_{o \sim D, z \sim p(\cdot \mid m), a \sim \pi_\theta(\cdot \mid o, \phi(o,z))}\left[\log \frac{\pi_{\bar{\theta}}(a \mid o, \phi(o,z))}{\mathbb{E}_{z' \sim p(\cdot \mid m), g' = \phi(o,z')}\left[\pi_{\bar{\theta}}(a \mid o, g')\right]}\right].
\end{aligned}
$$

Similarly, we have for the second mutual information term that

$$
\begin{aligned}
\mathcal{I}(m; g \mid o) &= \int p(m, o, g) \log \frac{p(m \mid o, g)}{p(m \mid o)} dm\, do\, dg \\
&= \mathbb{E}_{m,o,g}[\log \frac{p(m \mid o, g)}{p(m)}] \\
&= \mathbb{E}_{m,o \sim D}\left[\log p(m \mid o, g)\right] + \log|\mathcal{M}|.
\end{aligned}
$$

When maximizing the mutual information objectives described in Section 6.2, the Gumbel-softmax trick (Jang et al., 2016) can be used for discrete z.

D Complete Pseudocode

We begin by describing the *overview* of the training and adaptation procedures of MRA in Algorithm 2 and 3, respectively. Notably, the main difference between Algorithm 1 in the main text and Algorithm 2 here lies in that the latter is instantiated from the former, using the policy gradient and mutual information objectives depicted in Section 6.

Algorithm 2 MRA: Training in the MG set $\mathcal{M}$ (overview)

```
while not converged do
  for MG $m \in \mathcal{M}$ do
    Sample lower-level latent $z \sim p_\psi(\cdot|m)$
    Execute action $a \sim \pi_\theta(\cdot|o, g)$, where $g = \phi(o, z)$
    Push $(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{o}', \boldsymbol{g}, \boldsymbol{r})$ to replay buffer
    for $k = 1, \dots, K$ do
      Update critic $\zeta$ by minimizing Equation 8
      update policy at the $k$-th step $\theta_k$ by Equation 9
    end for
    Update $\theta$ by $\theta \leftarrow \theta + \alpha(\theta_K - \theta)$
    Update $\phi$ to maximize the RHS of Equation 11;
    Update $\psi$ and $\xi$ by Equation 13
    Update delayed parameters $\bar{\theta}$ and $\bar{\zeta}$
  end for
end while
```

Algorithm 3 MRA: Adaptation in an evaluation Markov Game $m' \in \mathcal{M}'$ (overview)

```
while not converged do
  Sample lower-level latent $z \sim p_\psi(\cdot|m')$
  Execute action $a \sim \pi_\theta(\cdot|o, g)$, where $g = \phi(o, z)$
  Push $(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{o}', \boldsymbol{r})$ to replay buffer
  Update critic $\zeta$ by minimizing Equation 8
  Update $\theta$ and $\phi$ by Equation 10
  Update delayed parameters $\bar{\theta}$ and $\bar{\zeta}$
end while
```

The complete pseudocode of the training and adaptation procedures of MRA that contains the training details and agent indexes is provided in Algorithm 4 and Algorithm 5, respectively.

Algorithm 4 MRA: Training Procedure of MRA (complete)

```
Input: Training set $\mathcal{M}$ that contains Markov Games constructed by varying the population (i.e., the number of agents) from the same underlying environment.
Initialize $P$ threads of games
Initialize $T_{\text{update}} \leftarrow 0$
Initialize replay buffer $\mathcal{D}_m$ for each Markov Game $m$
while total episode number not reach do
  for Markov Game $m = 1, \dots, |\mathcal{M}|$ do
    Reset game, each agent $i$ samples lower-level latent code $z^i \sim p_\psi(z|m)$
    for time steps in an episode do
      Each agent $i$ executes action $a^i \sim \pi\left(\cdot|o^i, \phi_i(o^i, z^i); \theta^i\right)$ simultaneously and get reward $r^i$, next observation $o'^i$
      Push $(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{o'}, \boldsymbol{g}, \boldsymbol{r})$ to replay buffer $\mathcal{D}_m$
      $\boldsymbol{o} \leftarrow \boldsymbol{o'}$
      $T_{\text{update}} \leftarrow T_{\text{update}} + P$
      if $T_{\text{update}}\%(\text{min steps per update}) \leq P$ then
        A mini-batch of $B$ samples of $(\boldsymbol{o}_b, \boldsymbol{a}_b, \boldsymbol{o'}_b, \boldsymbol{g}_b, \boldsymbol{r}_b)$ is sampled from $\mathcal{D}_m$
        for $k = 1$ to $K$ do
          Update all agents' critic parameter $\zeta^i$ by minimizing $\mathcal{L}\left(\zeta^i\right) = \frac{1}{B}\sum_b \left(\mathcal{B}^i_{\boldsymbol{\pi}} Q - Q\left(\boldsymbol{o}_b, \boldsymbol{a}_b, g^i_b; \zeta^i\right)\right)^2$,
                where $\mathcal{B}^i_{\boldsymbol{\pi}} Q = r^i_b + \gamma \mathbb{E}_{\boldsymbol{a'} \sim \bar{\boldsymbol{\pi}}}\left[Q\left(\boldsymbol{o'_b}, \boldsymbol{a'}, g^i_b; \bar{\zeta}^i\right)\right]$
          The $k$-th step of policy parameter $\theta^i_k$ is updated by gradient ascent:
                $\nabla_{\theta^i} J = \frac{1}{B}\sum_b \nabla_{\theta^i} \log \pi\left(a^i|o^i_b, g^i_b; \theta^i_k\right) Q\left(\boldsymbol{o}_b, \boldsymbol{a}_b, g^i_b; \zeta^i\right)$
        end for
        Update all agents' $\theta^i$ by $\theta^i \leftarrow \theta^i + \alpha(\theta^i_K - \theta^i)$
        Sample $n$ latent code $z'^i$ and approximate $p(a|o^i_b)$ by $\frac{1}{n}\sum \pi\left(a^i|o^i_b, \phi^i\left(o^i_b, z'^i\right); \bar{\theta}^i\right)$
        Update all agents' $\phi_i$ by maximizing:
              $\mathcal{L}\left(\phi^i\right) = \frac{1}{B}\sum_b \mathbb{E}_{a^i \sim \pi(\cdot|o^i_b, \phi^i(o^i_b, z^i); \theta^i)}\left[\log\left(\pi(a^i|o^i_b, \phi^i(o^i_b, z^i); \bar{\theta}^i)/p(a|o)\right)\right]$
        Update $\psi$ and auxiliary network $\xi$ simultaneously by minimizing:
              $\mathcal{L}\left(\psi, \xi\right) = \mathbb{E}_{z'^i \sim p_\psi(\cdot|m)}\left[-y \log\left(p\left(\hat{y}|o^i_b, \phi(o^i_b, z'^i); \xi\right)\right)\right]$
        Update all agents' delayed parameters $\bar{\theta}^i$ and $\bar{\zeta}^i$
      end if
    end for
  end for
end while
Output: Parameter $\psi$ and $\theta^i$, $\phi^i$, $\zeta^i$ for each agent $i$.
```

Algorithm 5 MRA: Adaptation Procedure of MRA (complete)

```
Input: Trained parameters from Algorithm 4, an evaluation Markov Game $m' \in \mathcal{M}'$.
Initialize replay buffer $\mathcal{D}$;
Each agent $i$ samples lower-level latent code $z^i \sim p_\psi(z|m')$;
while total episode number not reach do
    Reset game and receive initial observation $\boldsymbol{o}$;
    for time steps in an episode do
        Each agent $i$ executes action $a^i \sim \pi\left(\cdot|o^i, \phi_i(o^i, z^i); \theta^i\right)$ simultaneously and get reward $r^i$, next observation $o'^i$
        Push $(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{o'}, \boldsymbol{r})$ to replay buffer $\mathcal{D}_{m'}$
        $\boldsymbol{o} \leftarrow \boldsymbol{o'}$
        A mini-batch of $B$ samples of $(\boldsymbol{o}_b, \boldsymbol{a}_b, \boldsymbol{o'}_b, \boldsymbol{r}_b)$ is sampled from $\mathcal{D}_m$
        Calculate the detached relational graph $g^i = \phi_i(o^i_b, z^i_b)$
        Update all agents' critic parameter $\zeta^i$ by minimizing:
            $\mathcal{L}\left(\zeta^i\right) = \frac{1}{B}\sum_b \left(\mathcal{B}^i_{\boldsymbol{\pi}} Q - Q\left(\boldsymbol{o}_b, \boldsymbol{a}_b, g^i; \zeta^i\right)\right)^2$, where $\mathcal{B}^i_{\boldsymbol{\pi}} Q = r^i_b + \gamma \mathbb{E}_{\boldsymbol{a'} \sim \bar{\boldsymbol{\pi}}}\left[Q\left(\boldsymbol{o'_b}, \boldsymbol{a'}, g^i; \bar{\zeta}^i\right)\right]$
        Update all agents' parameter $\omega^i = (\theta^i, \phi^i)$ by gradient ascent:
            $\nabla_{\omega^i} J = \frac{1}{B}\sum_b \nabla_{\omega^i} \log \pi\left(a^i|o^i_b, \phi^i(o^i_b, z^i_b); \theta^i_k\right) Q\left(\boldsymbol{o}_b, \boldsymbol{a}_b, g^i; \zeta^i\right)$
        Update all agents' delayed parameters $\bar{\theta}^i$ and $\bar{\zeta}^i$
    end for
end while
Output: Parameter $\psi$ and $\theta^i$, $\phi^i$, $\zeta^i$ for each agent $i$.
```

E Details of Experiments

E.1 Experiment Settings and Task Descriptions

Treasure Collection: Each agent is with the goal to collect more treasures in an episode. Treasures disappear and re-generate at a random location when touched by agents. Agents must act according to the distances to other agents and treasures to gain high rewards, and paying attention to the right agents is key to success in this task.

Resource occupation: Agents receive rewards for occupying varisized resource landmarks: higher reward if one agent is occupying a larger resource with fewer other agents in it. Intuitively, training in various MGs could improve the performance in each single game benefiting from knowledge transfer. Specifically, agents can learn to move to large resources in small-population games, and learn to keep away from other agents in the competitive games with large populations. We will verify the benefit of such knowledge transfer with meta representations in Section 7.1.

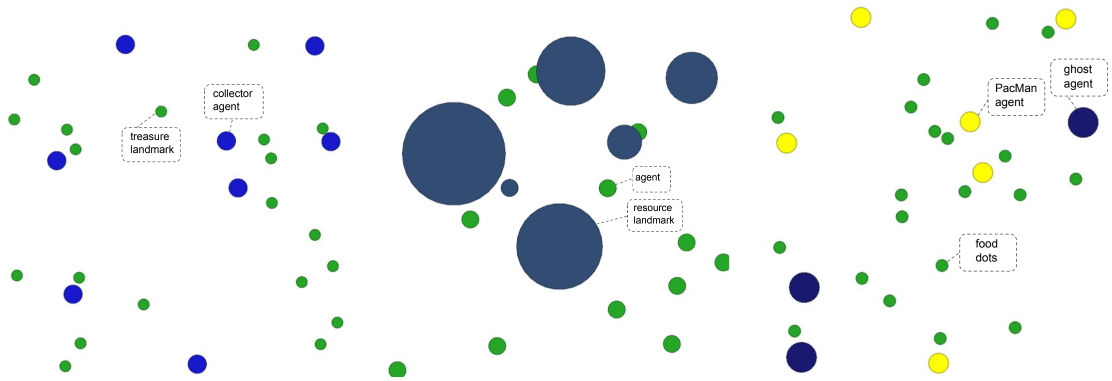
Pacman-like World: There are competing agents in the Pacman-like World, Pac-Man agents and ghost agents. Pac-Man agents are with goals to collect food and elude ghost agents, and ghost agents are with goals to touch the Pac-Man agents. The environment is similar to the predatory-prey environment, but with additional food dots, which means that the Pacman game is not a zero-sum game.

The screenshots of the three environments in the experiments are shown in Figure 6. The treasures in the treasure collection environment and the food dots in the Pacman-like world are randomly initialized in the position range of $[-1, +1]$, and regenerated when touched by collector agents and PacMan agents, respectively. For an n-resource occupation task, the sizes of the resource landmarks are pre-defined as $\{0.1, 0.2, \ldots, 0.1 \times n\}$ and fixed in each episode.

We adopt the same set of hyperparameters for experiments. Specifically, 12 rollouts are executed in parallel when training. The maximum length of the replay buffer is $1e6$. The episode length is set to 20 and the number of random seeds is set to 4. The dimension of the latent code z is 6. The critic also adopts a self-attention network in a similar way with MAAC (Iqbal & Sha, 2018). And the number of gradient steps of policy and critic parameters in each update, i.e., K, is set as 10. And $\alpha = 1$ works well in experiments. Batch size is set to 1024 and Adam is used as the optimizer. The initial learning rate is set to 0.0003. In all experiments, we use one NVIDIA Tesla P40 GPU.

E.2 Cross-Comparison Results

We provide the cross-comparison results in the PacMan-like world. The comparisons are conducted between the MRA agents trained in multiple MGs and the agents trained in a single MG. The score is summed in each episode, averaged across homogeneous agents on 40 runs, and normalized.



(a) Treasure collection. (b) Resource occupation. (c) Pacman-like world.

Figure 6: The illustration of the three environments that are used in our experiments.

Table 1: PacMan scores.

PacMan Ghosts	Single	MRA
Single	0.78	**1.00**
MRA	0.54	0.89

Table 2: Ghost scores.

PacMan Ghosts	Single	MRA
Single	0.82	0.59
MRA	**1.00**	0.85

The cross-comparison results are shown in Table 1 and Table 2. We can see that the agents created by the proposed MRA outperform the single-MG counterparts for both PacMan agents and ghosts agents, validating the effectiveness of the proposed method.

E.3 Ablation on the Implementation Variants

We now depict two variants of implementing $\phi(o, z)$ and compare them with the default implementation, which we denote as the option (Sutton et al., 1999) architecture.

Specifically, the variants we consider are the ones discussed in (Florensa et al., 2017). The first variant is to concatenate z to each entity of the observation decomposition $o^i = \left[o^i_s, o^i_1, \cdots, o^i_j, \cdots, o^i_N\right]$. The same relational representation is also adopted to generate the relational graph g. The second variant is to perform the outer product between each observation entity and z. We refer to the two variants as "concat" and "bilinear", respectively.

The performance of the three implementations is evaluated in the 6-resource occupation environment. The number of training MGs is set to 3, and the numbers of agents in the 3 games are $\{6, 9, 12\}$. The results in Figure 7 show that all the three implementations can obtain agents that effectively act in all the 3 scenarios. And the default option architecture achieves better performance than the other two variants. The possible reason is that the lower-level latent code z in the option architecture can explicitly control the structural factors and can thus learn the common knowledge more quickly and better.

E.4 Ablation Study on the Size of Training MG Set

The information in all the training MGs determines the common knowledge that agents can learn. We provide ablation study on the number of training MGs. In the resource occupation environment, we train the agents in three settings, each of which is with different size of training MG set: 2, 3 and 4. Specifically, the population size of the three settings are: $\{6, 12\}$, $\{6, 9, 12\}$ and $\{6, 9, 12, 15\}$. The curves in MGs with $\{6, 9, 12\}$ populations are shown in Figure 8.

We observe that when the size of the training MG set is greater than 2, the benefits of the meta-representation are obvious. The knowledge that agents learn from few training MGs, e.g., 1 or 2 training MGs, is limited, and the random exploration bottleneck still exists. However, the performance can be significantly improved by leveraging the information from more training MGs, e.g., 3 or 4 training MGs, where the common knowledge is more likely to be distilled and thus guiding the exploration.

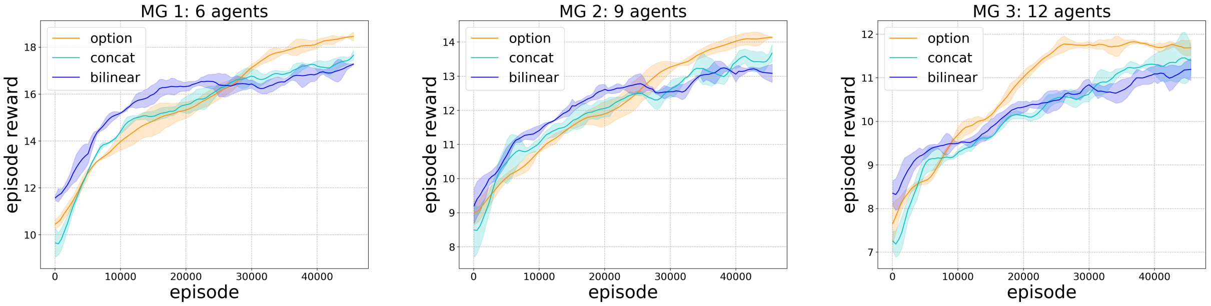


Figure 7: Performances of different implementation variants in the resource occupation environment.

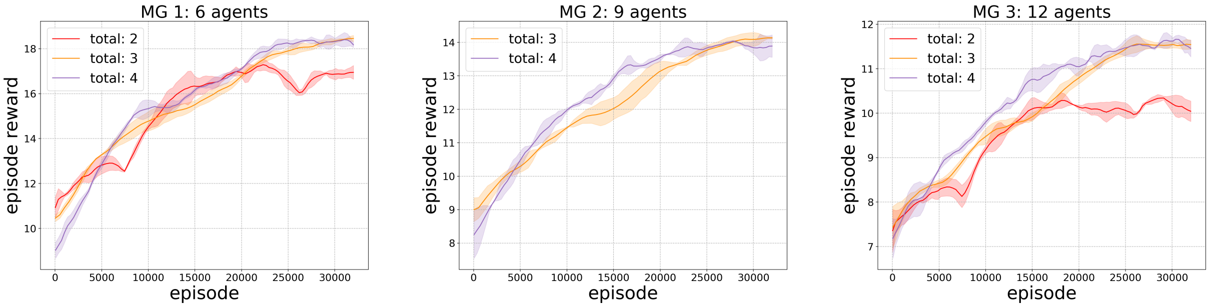
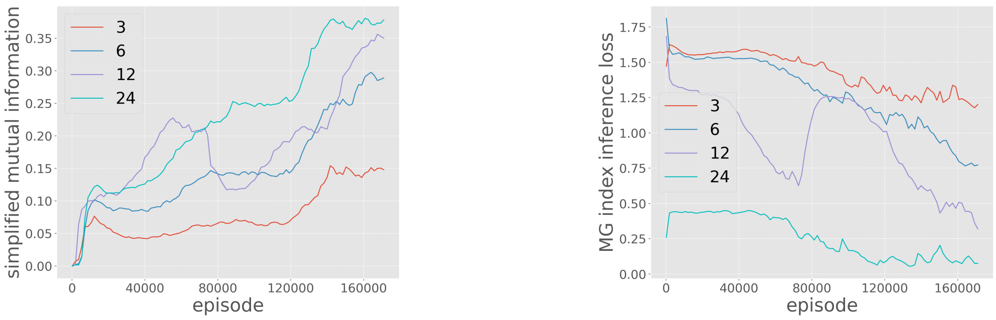


Figure 8: Ablation study on the size of the training MG set in the resource occupation environment.

E.5 Training Curves

We provide the training phase curves of the approximated mutual information $I(g; a \mid o)$ and the inference loss of MG index output by auxiliary network ξ. The curves are shown in Figure 9.



(a) Curve of mutual information $I(g; a \mid o)$ during training.

(b) The loss curve of the MG index inference.

Figure 9: Curves during training. The total number of training MGs is 4, with $\{3, 6, 12, 24\}$ agents in the resource occupation environment.

E.6 Visualizations

We visualize the trajectories of one agent in a resource occupation game in Figure 10. Green dots and blue dots are agents and resources, respectively. When agents are only trained in this MG with different random seeds, different behaviors are obtained. This indicates that agents trained in single MGs are confined to environmental settings. Agents only learn the best responses and fit an NE. However, if the agents are only aware of some of the successful behaviors, the generalization will be constrained. On the contrary, the proposed MRA algorithm has a large capacity to represent multiple strategies by incorporating different relational graph with the distilled common knowledge, which leads to various modes of behaviors that are reasonable and diverse.

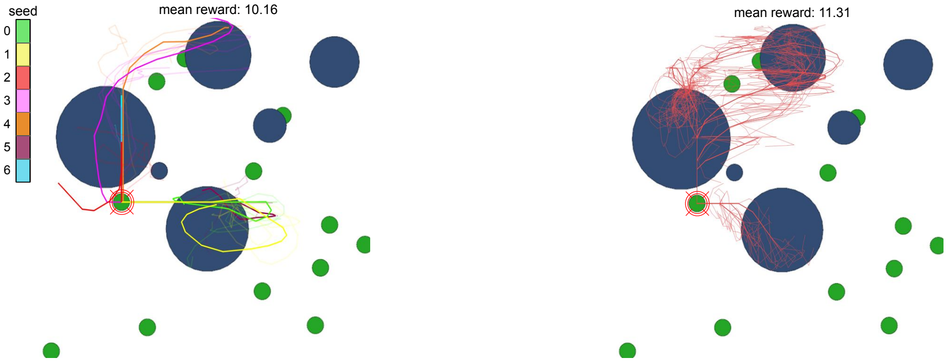


Figure 10: Trajectory visualization in the resource occupation environment. **Left:** Trajectories of agents that are trained in a single Markov Game. **Right:** Trajectories of agents that are trained in multiple Markov Games.

We also visualize some instances of the learned relation variations, i.e., different relational graph g under observation o, as well as how agents make the smartest decisions under different variations in Fig. 11. The common knowledge learned by the agent can be interpreted as "moving to less-agent resources". Specifically, in Fig. 11(a) the black agent makes decisions to move left by focusing on the topmost red agents which are occupying a resource. By focusing on the leftmost red agents in Fig. 11(b), the black agent makes decisions to move up. Although such behavior might not be optimal, since the topmost resource is smaller than the leftmost resource, this variation helps agents learn common knowledge and optimally behave in an unseen MG by incorporating the optimal relation mapping in that game.

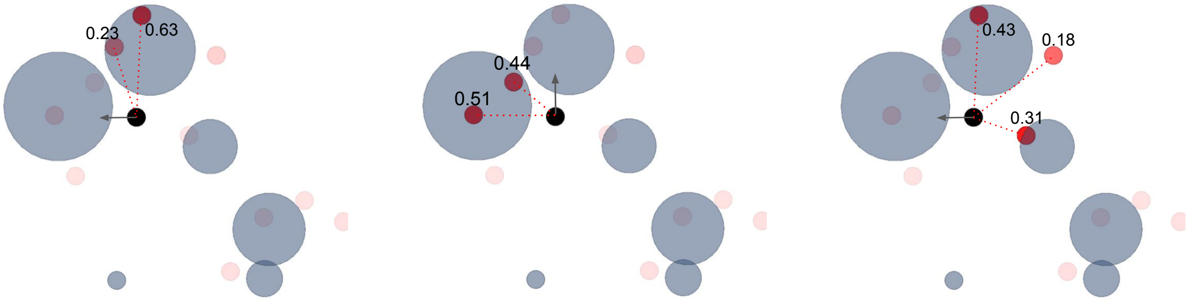


Figure 11: Instances of different learned relational graph and the corresponding actions that agents take. We visualize how the black agent makes different reasonable decisions by incorporating different relational graphs. The relation scores $g^{i,j}$ that are smaller than 0.1 are not shown.