
Role Play: Learning Adaptive Role-Specific Strategies in Multi-Agent Interactions

Weifan Long¹, Wen Wen², Peng Zhai^{1,3}, Lihua Zhang^{1,3*}
wflong21@m.fudan.edu.cn, w_wen1113@tju.edu.cn,
pzhai_lihuazhang@fudan.edu.cn

¹Academy for Engineering and Technology, Fudan University, Shanghai 200433, China

²School of New Media and Communication, Tianjin University, Tianjin 300072, China

³Ji Hua Laboratory, Foshan 528251, China

Abstract

Zero-shot coordination problem in multi-agent reinforcement learning (MARL), which requires agents to adapt to unseen agents, has attracted increasing attention. Traditional approaches often rely on the Self-Play (SP) framework to generate a diverse set of policies in a policy pool, which serves to improve the generalization capability of the final agent. However, these frameworks may struggle to capture the full spectrum of potential strategies, especially in real-world scenarios that demand agents balance cooperation with competition. In such settings, agents need strategies that can adapt to varying and often conflicting goals. Drawing inspiration from Social Value Orientation (SVO) where individuals maintain stable value orientations during interactions with others we propose a novel framework called *Role Play* (RP). RP employs role embeddings to transform the challenge of policy diversity into a more manageable diversity of roles. It trains a common policy with role embeddings observation and employ a role predictor to estimate the joint role embeddings of other agents, helping the learning agent adapt to its assigned role. We theoretically prove that an approximate optimal policy can be achieved by optimizing the expected cumulative reward relative to an approximate role-based policy. Experimental results in both cooperative (*Overcooked*) and mixed-motive games (*Harvest*, *CleanUp*) reveal that RP consistently outperforms strong baselines when interacting with unseen agents, highlighting its robustness and adaptability in complex environments.

1 Introduction

Artificial Intelligence (AI) has achieved remarkable success in mastering a wide range of strategic and competitive games, as demonstrated by notable research in this area (Silver et al., 2018; Vinyals et al., 2019; Berner et al., 2019). Significant advancements have also been made in cooperative settings, where agents are trained to collaborate with either humans or other agents to achieve common goals (Carroll et al., 2019; Zhao et al., 2023). However, in real-world applications such as autonomous driving, interactions among agents often display mixed motives, combining elements of both cooperation and competition (Schwarting et al., 2019). In these mixed-motive environments, agents face complex interactions where each participant has distinct objectives. For example, in a public goods game, agents must carefully balance the benefits of contributing to a collective resource against the costs of their individual contributions (Gibbons et al., 1992). These environments present

*Corresponding Author

significant challenges, requiring agents to develop sophisticated adaptation strategies to effectively interact with others who have varying incentives.

Zero-shot coordination is well-recognized in the context of multi-agent reinforcement learning (MARL), particularly for agents that need to interact effectively with new partners they have not encountered during training (Hu et al., 2020). Self play (SP) is an effective framework for this challenge (Lucas and Allen, 2022; Lupu et al., 2021; Zhao et al., 2023). The SP-based frameworks typically build a policy pool through SP, utilized to enhance the generalization capabilities of the final agent. Various techniques aim to increase the diversity of this policy pool to improve the agent’s ability to generalize across different scenarios (Garnelo et al., 2021; Zhao et al., 2023). In mixed-motive games, there is a greater need for policy diversity to adapt to the varying and often conflicting goals resulting from the imperfect alignment of incentives among group members. However, the policy pool, primarily composed of past iterations of policies from a given population, captures only a limited range of the policy space. This limitation can prevent agents from effectively managing novel situations or policies not previously encountered in the training set.

Unlike existing policy pool based works, our approach try to develop a general model which can generate policies with different value orientation. Given the inherent challenges in representing policies directly due to their complexity, we propose projecting the policy space into a more compact dimension. Inspired by Social Value Orientation (SVO) (McKee et al., 2020), in which individuals maintain stable value orientations (roles) in interactions with others, we proposed *Role Play* (RP), which compress the vast MARL policy space into a more manageable “human role space.” This simplification aims to improve both the interpretability and efficiency of agent interactions. Furthermore, drawing on social intuition (Lieberman, 2000; Jellema et al., 2024) that humans estimate the behaviors of others during interactions to make better decisions, we introduce a role predictor to estimate the joint role embeddings of other agents, aiding the learning agent in adapting to its assigned role. This setup enables agents to learn and adapt to their assigned roles more effectively, enhancing their performance across various interactive scenarios.

In this work, we introduce a novel framework, *Role Play* (RP), specifically designed to address the zero-shot coordination problem in multi-agent interactions. Our approach is distinguished by several key innovations:

- **Role Embedding:** We utilize a sophisticated reward mapping function to project the extensive policy space into a more manageable role embedding space. This transformation facilitates structured and strategic navigation through the complex landscape of agent behaviors. We theoretically prove that an approximate optimal policy can be obtained by optimizing the expected cumulative reward with respect to an approximate role-based policy.
- **Role Predictor:** Inspired by social intuition, we have developed a role predictor that estimates the joint role embeddings of other agents. This module enhances the agent’s ability to accurately predict and adapt to the role-based policies of other agents, enabling the learning agent to adapt more effectively to its assigned role.
- **Meta-task Learning:** We employ meta-learning techniques to model agent interactions as meta-tasks, which allows the learning agent to extrapolate from limited experiences to new, unseen scenarios. This approach significantly improve the adaptability of the learning agent to different roles and strategies.

These innovations collectively enhance the capability of agents to adapt and perform in complex multi-agent environments, establishing RP as a robust solution to zero-shot coordination challenges in MARL. To gain a deeper understanding of our framework and explore additional visualizations, we invite readers to visit our project website, where more detailed results are provided².

2 Related works

A significant body of research focuses on enhancing the zero-shot coordination of MARL agents when interacting with unfamiliar partners (Kirk et al., 2021). Most existing methods that aim to improve generalization across diverse agent interactions utilize the SP framework (Lanctot et al., 2017; Bai et al., 2020), typically employing a policy pool to train agents in adapting to varied

²https://weifan408.github.io/role_play_web/

strategies. FCP (Strouse et al., 2021) introduces a two-stage framework that initially generates a pool of self-play policies and their prior versions, followed by training an adaptive policy against this pool. Additionally, some approaches improves the diversity of within the policy pool to cultivate more robust adaptive policies (Garnelo et al., 2021; Zhao et al., 2023). TrajeDi (Lupu et al., 2021) applies the Jensen-Shannon Divergence across policies to foster the training of diverse strategies. AnyPlay (Lucas and Allen, 2022) introduces an auxiliary loss function and intrinsic rewards to aid self-play-based agents in generalizing to more complex scenarios of both intra-algorithm and inter-algorithm cross-play. Furthermore, BRDiv (Rahman et al., 2023) assesses diversity based on the compatibility of teammate policies in terms of returns. HSP (Yu et al., 2023) employs event-based reward shaping to enhance the diversity of the policy pool. While these SP-based methods have achieved success, the reliance on a policy pool can be limiting. The policy pool, primarily composed of past iterations of policies from a given population, captures only a limited range of the policy space. This limitation can prevent agents from effectively managing novel situations or policies not previously encountered during training. Our approach diverges from the conventional use of a policy pool. Instead, we aim to learn a versatile policy capable of generating a spectrum of behaviors through role embeddings. This allows for broader adaptation across various strategic scenarios, enhancing the overall diversity and robustness of policies.

Our approach enhances the learning agent’s adaptability by predicting the roles of other agents. Opponent modeling is traditional method which characterizing the behaviors, goals, or beliefs of opponents, enabling agents to adjust their policies to effectively adapt to various opponents. ToMnet (Rabinowitz et al., 2018) aims to equip agents with a theory of mind analogous to that of humans. Similarly, SOM (Raileanu et al., 2018) adopts a unique approach to theory of mind by utilizing the agent’s own policy to infer the opponent’s goals. LOLA (Foerster et al., 2017) explicitly considers how an agent’s policy might influence future parameter updates of its opponent, effectively anticipating and shaping the opponent’s learning trajectory. Meta-MAPG (Kim et al., 2021) integrates meta-learning to develop policies that can adapt to the learning process of opponents. Additionally, M-FOS (Lu et al., 2022) employs generic model-free optimization methods to learn meta-policies that are effective in long-horizon opponent shaping. While these methods focus on the learning dynamics of opponents, they tend to pay less attention to enhancing the generalization capabilities across diverse agent interactions. In contrast, our approach emphasizes the role-based adaptation, enabling agents to effectively engage with a wide range of roles and strategies without relying on detailed models of opponent learning dynamics.

We introduce meta-learning techniques to enable our policy leverage prior experiences across different roles, allowing it to quickly adapt to new tasks with minimal data or training (Beck et al., 2023). There are two canonical meta reinforcement learning algorithms that have been widely adopted in the literature (Song et al., 2019; Tack et al., 2022), Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) which uses meta-gradients, and Fast RL via Slow RL (RL²) (Duan et al., 2016), which uses which uses a history-dependent policy. MAML enables models to quickly adapt to new tasks with only a few gradient updates by optimizing the model’s parameters such that they can be fine-tuned efficiently for new tasks, without requiring major adjustments to the model architecture itself. RL² uses a slower, more deliberate reinforcement learning (RL) process to guide and optimize a faster learning agent; the slow RL agent captures long-term strategies and knowledge, which are then leveraged by the fast RL agent to quickly adapt to new tasks or environments. These methods have been successfully applied in various scenarios (Yu et al., 2020; Mitchell et al., 2021). In our work, we introduce meta-learning techniques to model the interactions between agents as meta-tasks, enabling the learning agent to generalize from a limited set of experiences to new, unseen agents and roles. This approach enhances the adaptability of the learning agent to different roles and strategies, enabling it to effectively navigate complex multi-agent environments.

3 Preliminary

In this work, we study a role-based Markov game, which is represented by $\mathcal{M} = \langle m, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \psi, z, \gamma \rangle$. Here, m denotes the number of agents, and \mathcal{S} represents a finite set of states. The joint action space is given by $\mathcal{A} = \prod_{i=1}^m A^i$, where A^i is the action space for agent i . The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ defines the probability of transitioning from one state to another given a joint action. Each agent i has a specific reward function $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, and the collection of these functions forms the joint reward function $\mathcal{R} = \{\mathcal{R}^i \mid i = 1, \dots, m\}$.

The joint observation function, $\mathcal{O} = \{\mathcal{O}^i \mid i = 1, \dots, m\}$, where each \mathcal{O}^i provides observations for agent i . We introduce a joint role embedding is represented as $\mathbf{z} = \{z^i \mid i = 1, \dots, n\}$, where each $z^i \in \mathbb{Z}$ is the role embedding for agent i , and \mathbb{Z} is the role embedding space. The reward feature mapping function $\psi(r^i, z^i)$ processes the reward r^i and the role embedding z^i to generate a new reward value for agent i . The discount factor is $\gamma \in [0, 1]$. Each agent i operates under the policy $\pi(a^i \mid o^i)$, where a^i is the action taken given the observation o^i . The joint policy of agent i with role embedding z^i is defined as $\pi(z^i)$, and the joint policy of all agents except agent i is denoted as $\pi^{-i}(\mathbf{z}^{-i}) = \prod_{j \neq i} \pi(z^j)$, where $\mathbf{z}^{-i} = \{z^j \mid j \neq i\}$ is the joint role embedding of all agents except agent i . Similarly, the joint observation of all agents except agent i is $\mathbf{o}^{-i} = \{o^j \mid j \neq i\}$, and the joint action of all agents except agent i is $\mathbf{a}^{-i} = \{a^j \mid j \neq i\}$.

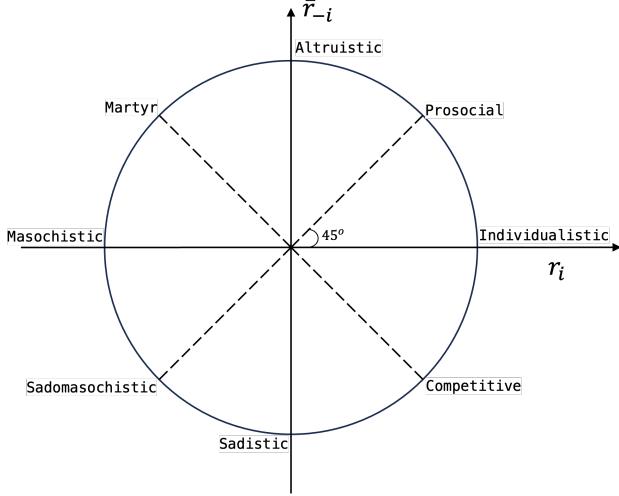


Figure 1: A graphical representation of the SVO framework.

We employ Social Value Orientation (SVO) (Murphy and Ackermann, 2014) to categorize agent roles in mixed-motive games. SVO is a psychological construct that captures individual differences in the value orientations during social interactions. It classifies individuals based on their preferences for outcomes. The ring formulation of SVO is illustrated in Fig. 1. The SVO angle of agent i , denoted as θ^i , is computed based on the agent's own reward and the average reward of other agents:

$$\theta^i = \arctan \left(\frac{r^i}{\bar{r}^{-i}} \right), \quad (1)$$

where r_i is the reward of agent i and \bar{r}^{-i} is the average reward of other agents. In general, SVO can be categorized into eight categories: *Masochistic*, *Sadomasochistic*, *Sadistic*, *Competitive*, *Individualistic*, *Prosocial*, *Altruistic* and *Martyr*, corresponding to $\theta^i = \frac{k\pi}{4}$ for $k = -4, \dots, 3$. We use the variant of SVO for reward shaping (Peng et al., 2021), which defines the SVO-shaped reward for agent i as:

$$r'_i = \cos(\theta^i) \cdot r_i + \sin(\theta^i) \cdot \bar{r}^{-i}, \quad (2)$$

where r'_i is SVO-shaped reward for agent i . When $\theta^i = 0$ (*Individualistic*), the reward is the same as the original reward r^i . When $\theta^i = \frac{\pi}{4}$ (*Prosocial*), the reward incorporates both the agent's reward and the average reward of others, effectively considering collective outcomes. When $\theta^i = \frac{\pi}{2}$ (*Altruistic*), the reward focuses solely on the average reward of the other agents.

4 Method

Role play (RP) is designed to train a single policy which can adapt all roles in role space, enabling agents to dynamically adapt their strategies based on their assigned roles. Different with existing policy-pool-based methods, RP transforms the policy diversity challenges into more manageable diversity of role representations. The overall framework is illustrated in Fig. 2.

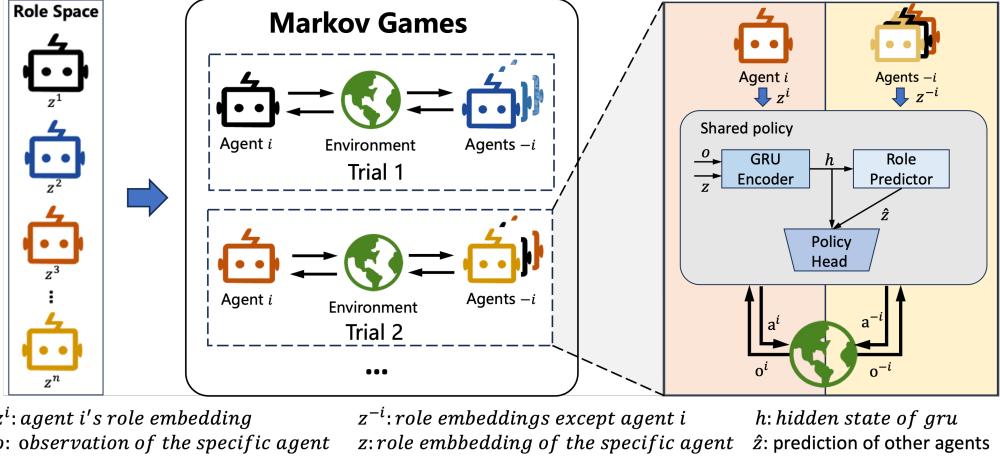


Figure 2: **Role play framework:** Different agents share the same policy network but with different role embedding inputs. The role policy prediction network is trained to predict the role embeddings of other agents. The learning agent optimizes its policy based on the predicted roles.

4.1 Role embedding

In RP framework, each agent i is assigned a randomly sampled role embedding z^i in every trial and learns to effectively adapt to and play this role. The challenge lies in defining the role space and enabling agents to adapt efficiently to their assigned roles. To address this, we employ a reward shaping method and introduce a reward feature mapping function $\psi(r^i, z^i)$ that processes the reward r^i and the role embedding z^i to generate a new reward value for agent i . This function $\psi(r^i, z^i)$ is specifically designed to capture the role-specific reward information, which is crucial for the agent to adapt to its role. During training, agents interact with a variety of other randomly sampled roles, optimizing their strategies based on these interactions. Consequently, if the role space is sufficiently diverse to adequately represent the entire policy space adequately, agents are equipped to effectively handle interactions with previously unseen agents.

Each agent i takes action without communications, based on its observation o^i and role embedding z^i , receiving individual rewards r^i . Its goal is to maximize its expected cumulative discount reward

$$J(\pi) = \mathbb{E}_{z^i, z^{-i} \sim z} [J(\pi(z^i), \pi(z^{-i}))], \quad (3)$$

where $J(\pi(z^i), \pi(z^{-i}))$ denotes the expected sum reward achieved by agent i using policy $\pi(z^i)$ while other agents use policies $\pi^{-i}(z^{-i})$. The detailed form of $J(\pi(z^i), \pi(z^{-i}))$ is given by

$$J(\pi(z^i), \pi(z^{-i})) = \mathbb{E}_{\substack{s_{t+1} \sim \mathcal{P}(\cdot | s_t, a_t^i, a_t^{-i}) \\ a_t^i \sim \pi^i(\cdot | o_t^i, z^i), a_t^{-i} \sim \pi^{-i}(\cdot | o^{-i}, z^{-i})}} \left[\sum_{t=0}^{\infty} \gamma^t \psi(\mathcal{R}(s_t, a_t^i, a_t^{-i}), z^i) \right]. \quad (4)$$

Given the vast size of the policy space, it is challenging for the mapped role space \mathbb{Z} to fully represent the entire policy space one-to-one. Inevitably, this reduction results in some loss of information. To address this issue, we analyze the adaptability with respect to missing strategies using an ϵ -close (Ko, 2006; Zhao et al., 2023) approximation.

Definition 4.1. (Zhao et al., 2023) We define that a random policy π' is ϵ -close to policy $\pi(z')$ at observation o_t if

$$\left| \frac{\pi'(a'_t | o'_t)}{\pi(a'_t | o'_t, z')} - 1 \right| < \epsilon \quad (5)$$

for all $a'_t \in \mathcal{A}$. If this condition is satisfied at every $o'_t \in \mathcal{O}$, we call π' is ϵ -close to $\pi(z')$.

Based on Definition 4.1, if the random policy π' is ϵ -close to the role policy $\pi(z')$, we can derive the following theorem.

Theorem 4.1. For a finite MDP with T time steps and a specific role policy $\pi(z)$, if any random policy π' is ϵ -close to the role policy $\pi(z')$, then we have

$$\left| \frac{J(\pi(z), \pi')}{J(\pi(z), \pi(z'))} - 1 \right| \leq \epsilon T. \quad (6)$$

Proof. To maintain the readability of the paper, the proof is included in the A. \square

Theorem 4.1 suggests if a random policy π' is ϵ -close to the role policy $\pi(z')$, the expected cumulative reward for the learning policy $\pi(z)$ in relation to the random policy π' is approximately the same as optimizing it in relation to the role policy $\pi(z')$. This implies that the learning agent can obtain an approximately optimal policy by optimizing the expected cumulative reward with respect to a role-based policy. However, multi-agent games are complex and dynamic, where each agent's actions can unpredictably influence others. This uncertainty makes it challenging for agents to optimize their behavior effectively. To address this issue, we propose a prediction method to estimate the ϵ -close role policy.

4.2 Role predictor

We aim for the learning agent to accurately predict the joint role embeddings of other agents. To achieve this, we introduce a role predictor q_ϕ , which is trained to predict the joint role embeddings of other agents, denoted \hat{z}_t^{-i} , based on the history of the observation $o_{0:t}$ and its own role embedding z^i at time step t . Specifically, the prediction model is formulated as:

$$\hat{z}_t^{-i} = \arg \max_{\mathbf{z}^{-i}} q_\phi(\mathbf{z}^{-i} | o_{0:t}, z^i). \quad (7)$$

This model architecture enables the agent to integrate observational history and predefined role to infer the joint role of other agents within the environment, enhancing its predictive accuracy and adaptability.

Utilizing the role predictor q_ϕ and theorem 4.1, the learning agent optimizes its policy with the role embedding z^i by maximizing its expected cumulative discount reward

$$J(\pi(z^i), \pi(\mathbf{z}^{-i})) \approx J(\pi(z^i), \hat{\mathbf{z}}^{-i}) = \mathbb{E}_{\substack{s_{t+1} \sim \mathcal{P}(\cdot | s_t, a_t, a_t^{-i}) \\ a_t \sim \pi(\cdot | o_t, z^i, \hat{\mathbf{z}}^{-i})}} \left[\sum_{t=0}^{\infty} \gamma^t \psi(\mathcal{R}(s_t, a_t, a'_t), z^i) \right]. \quad (8)$$

Once the learning agent is assigned a role, it approximates the dynamics of the MARL problem into a unified single-agent RL framework and optimizes its policy independently.

Up to this point, the learning agent can adaptively interact with any unseen policies for a given role, assuming that both the policy and the role predictor are effectively trained. Unfortunately, the vast space of joint policies from other agents presents significant challenges in terms of generalization and stability.

4.3 Modeling role interactions as meta-tasks

To address the challenges of generalization and stability highlighted in the preceding discussion, we introduce meta-learning techniques (Duan et al., 2016; Finn et al., 2017; Fakoor et al., 2019) into RP. Meta-learning enables agents to generalize from a limited set of experiences to new, unseen scenarios effectively. In our approach, we model the interactions between the learning agent and other agents as meta-tasks, utilizing meta-learning techniques to develop the learning agent's policy.

We treat the role embeddings of other agents as context variables that encapsulate relevant information about the task environment. These role embeddings provide the contextual backdrop for the meta-learning process, offering a rich representation of the dynamic interactions within the multi-agent system. The sampling method of role embedding generates a probability distribution over tasks. Thus, we view the entire learning process as meta-learning tasks, where the agent must adapt to various roles in the environment, guided by the context provided by the role embeddings.

Algorithm 1 Role Play

```
1: Initialization:
2: Initialize role space  $\mathbb{Z}$ 
3: Initialize reward feature mapping function  $\psi$ 
4: Interaction:
5: for each iteration do
6:   for each trail do
7:     Sample role embeddings for all agents  $z^1, \dots, z^m$ 
8:     for each time step  $t$  do
9:       for each agent  $i$  do
10:        Predict the joint role embeddings of other agents  $\hat{z}_t^{-i} \sim q_\phi(\hat{z}_t^{-i} | o_{0:t}, z^i)$ 
11:        Sample actions  $a_t^i \sim \pi(a_t^i | o_t^i, z^i, \hat{z}_t^{-i})$ 
12:        Observe next observation  $o_{t+1}$  and rewards  $r_t^i$ 
13:         $r_t^i \leftarrow \psi(r_t^i, z^i)$ 
14:      end for
15:    end for
16:  end for
17:  Update the role predictor  $q_\phi$ 
18:  Update the policy  $\pi$  by a meta learning algorithm
19: end for
```

4.4 Role play framework

In this subsection, we summarize the RP framework and provide detailed insights into its practical training phase. The algorithm is delineated in Algorithm 1.

In practical training, we employ the typically meta-learning algorithm, RL^2 (Duan et al., 2016), to optimize the policy, as illustrated in Fig. 3. Throughout the iterative process, the algorithm dynamically assigns role embeddings z^i to agents, enabling them to predict joint role embeddings of other agents \hat{z}_t^{-i} . This facilitates action selection a_t^i based on these roles and observed interactions. Rewards are subsequently mapped using the defined feature function ψ . The role predictor q_ϕ is updated by minimizing the prediction error between the predicted role embedding and the true role embedding. The learning agent is trained to maximize the expected cumulative mapped reward by interacting with other agents.

5 Experiments

In this study, we initially assess the effectiveness of RP using the cooperative two-player game Overcooked (Carroll et al., 2019), which is the common benchmark for evaluating the collaboration ability of agents (Zhao et al., 2023; Yu et al., 2023; Rahman et al., 2023). We further assess its performance in two-player mixed-motive games, such as Harvest and CleanUp, where agents need to balance cooperation with competition. In these settings, agents must develop strategies that navigate conflicting goals to maximize both individual and collective outcomes. All subsequent internal analysis experiments are conducted within more complex mixed-motive environments. Ablation studies are conducted to verify the effectiveness of the role predictor and the meta-learning methods employed. Role behavior analysis is performed to evaluate the adaptability of the learning agent to different roles. Finally, we analyze performance of the role predictor in predicting the role embeddings of other agents.

Baselines. We compare the performance of our proposed RP method against several baseline strategies that represent a range of common and state-of-the-art approaches in zero-shot coordination challenge. These include *Trajectory Diversity* (*TrajeDi*) (Lupu et al., 2021), *AnyPlay* (Lucas and Allen, 2022), *Best-Response Diversity* (*BRDiv*) (Rahman et al., 2023) and *Hidden-Utility Self-Play* (*HSP*) (Yu et al., 2023). Each of these methods follows a two-stage process: generating a policy pool and then training a final policy. Notably, while all baseline methods train 16 policies in the first stage to ensure sufficient diversity and adaptability, **RP achieves comparable or superior performance by training a single policy in a streamlined, single-stage process**, significantly reducing both computational overhead and complexity.

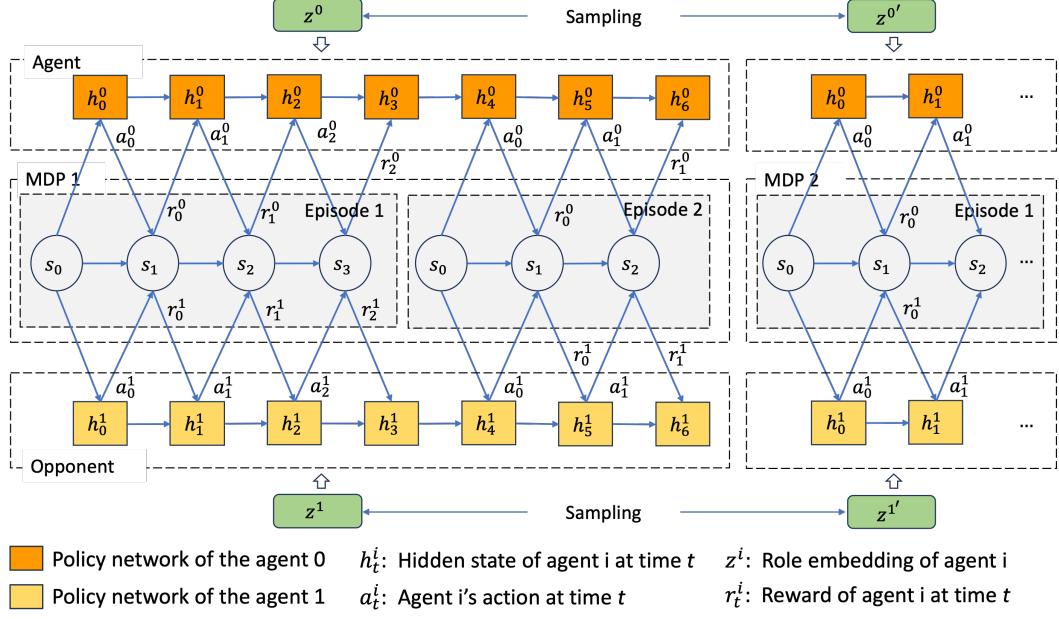


Figure 3: RL^2 based RP

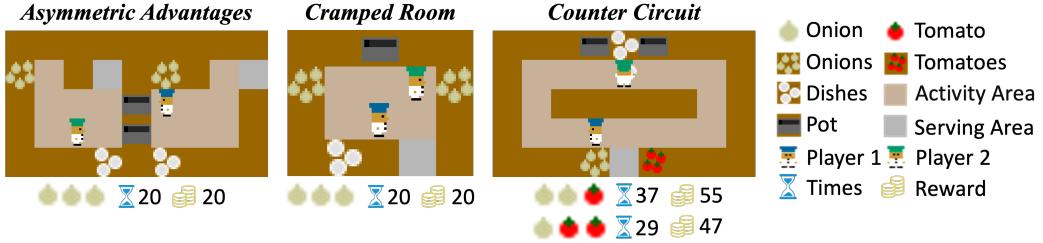


Figure 4: Layouts in Overcooked.

5.1 Zero Shot Results on Cooperative Games

Overcooked (Carroll et al., 2019) is a two-player cooperative game intended to test the collaboration ability of agents. In this game, agents work collaboratively to fulfill soup orders using ingredients like onions and tomatoes. Agents can move and interact with items, such as grabbing or serving soups, based on the game state. To complete an order, agents must combine the correct ingredients in a pot, cook them for a specified time, and then serve the soup with a dish to earn rewards. Each order has distinct cooking times and rewards.

Experimental setup. In Overcooked, we employ an event-based reward map function, which is a sparse reward function that rewards agents for some events happening. We select certain events already implemented in Overcooked and assign three preference levels-hate, neutral and like-for each event which will lead to a negative, zero and positive reward. The reward shaping function is defined as $\psi(r^i, z^i) = r^i + \sum_k (z_k^i * E_k)$, where $z_k^i \in [-1, 0, 1]$ is the preference of event k , and E_k is the reward of event k . Details of the experimental settings in Overcooked are provided in Appendix B.1.

Fig. 4 illustrates the three layouts considered: *Asymmetric Advantages*, *Cramped Room*, and *Counter Circuit*. The *Asymmetric Advantages* layout includes only onions, with two agents separated into distinct areas. *Cramped Room* also features an onion-only layout but places both agents in a shared activity space. *Counter Circuit* presents a more complex setup, with both onions and tomatoes; agents must collect ingredients and prepare the correct soup to earn rewards.

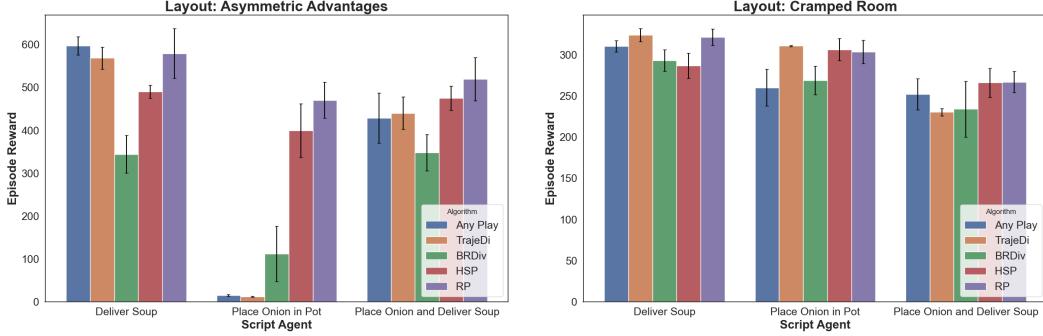


Figure 5: Zero-shot evaluation results with script agents on *Asymmetric Advantages* and *Cramped Room*.

We evaluated the performance of all algorithms in tasks where the trained agent interacts with various scripted agents implemented using HSP (Yu et al., 2023). In each task, policies engaged with the scripted agent over 100 episodes, and we calculated the mean and standard deviation of the episode rewards. Since the *Asymmetric Advantages* and *Cramped Room* layouts contain only onions, we assessed performance across the following tasks: *Deliver Soup Agent*, which the scripted agent only delivers soup; *Place Onion in Pot Agent*, which the agent only places the onion in the pot; and *Place Onion in Pot and Deliver Soup Agent*, which the agent performs both actions. Fig. 5 shows the evaluation results for these onion-only layouts. In the *Asymmetric Advantages* layout, AnyPlay and TrajeDi perform well in *Deliver Soup Agent* task but struggle in the *Place Onion in Pot Agent* task, as their policy pools from stage one rarely contain potting policies, which limits their adaptability. By contrast, both HSP and RP overcome this issue using reward shaping methods. While HSP is limited by the size of its policy pool, RP achieves higher rewards by generating a wider range of policies through role embeddings, allowing it to adaptively apply different reward shaping functions. In the simpler layout, *Cramped Room*, all algorithms achieve satisfactory performance. However, RP exhibits a higher level of adaptability by consistently adjusting its policies to suit varying tasks, outperforming others in maintaining robust results across different scripted agent interactions.

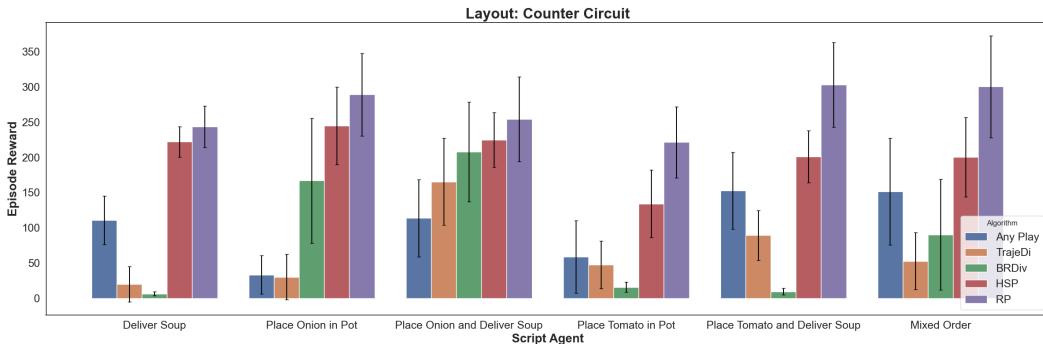


Figure 6: Zero-shot evaluation results with script agents on *Counter Circuit*.

In the more complex *Counter Circuit* layout, where agents must gather onions and tomatoes to prepare correct soup, we evaluated performance across three additional tasks: *Place Tomato in Pot Agent*, where the agent only places tomatoes in the pot; *Place Tomato in Pot and Deliver Soup Agent*, where the agent both places tomatoes in the pot and delivers soup; and *Mixed Order*, where the agent places various ingredients in the pot. The results are shown in Fig. 6. AnyPlay, TrajeDi and BRDiv obtain policy pools through an implicit way, which limits their ability to cover the diversity of strategies required in complex scenarios, resulting in poor performance across all tasks. In contrast, RP and HSP apply reward shaping methods (an explicit approach) to obtain a broader range of policies, leading to higher rewards. RP consistently outperforms HSP, highlighting its superior adaptability in challenging scenarios. Notably, some results exhibit significant standard

deviations, due to the sparse reward structure, where collecting or delivering orders variably impacts the total score.

5.2 Zero Shot Results on Mixed-Motive Games

In this subsection, we evaluate the performance of the RP framework in two-player mixed-motive games, *Harvest* and *CleanUp* (Hughes et al., 2018; Leibo et al., 2021), which require agents to balance individual contributions against collective benefits. These games introduce a higher level of complexity due to inherent conflicting interests, providing a challenging test environment for agents. To clarify the mixed-motive game settings, we provide detailed explanations of these games.

Harvest: agents face a *common-pool resource dilemma* where apple resources regenerate more effectively when left unharvested in groups. Agents can choose to harvest aggressively, risking future availability, or sustainably, by leaving some apples to promote long-term regeneration. Additionally, agents can use beams to penalize others, causing the penalized agent to temporarily disappear from the game for a few steps.

CleanUp: a *public goods game* in which apple growth in an orchard is hindered by rising pollution levels in a nearby river. When pollution is high, apple growth stops entirely. Agents can reduce pollution by leaving the orchard to work in polluted areas, highlighting the importance of individual efforts in maintaining shared resources. Agents are also able to use beams to penalize others.

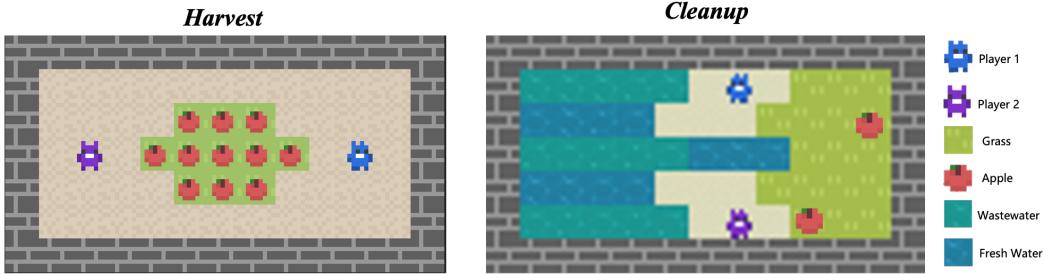


Figure 7: Harvest and CleanUp games.

Experimental setup. We implemented the scenarios using the MeltingPot framework (Leibo et al., 2021) and applied SVO (Murphy and Ackermann, 2014) to categorize agent roles. Utilizing the ring formulation of SVO, we introduced SVO-based (role-based) rewards as intrinsic motivators for agents to learn role-specific strategies. The reward feature mapping function ψ is defined as:

$$\psi(\mathbf{r}, z^i) = w * r^i + (1 - w) * \underbrace{\left[\cos(z^i)r^i + \sin(z^i)\bar{r}^{-i} \right]}_{\text{SVO-based reward shaping}}, \quad (9)$$

where r^i denotes the individual reward for agent i , \bar{r}^{-i} the average reward of other agents, and w a hyperparameter adjusting the impact of role-based rewards. We use eight main roles $\{z^i = \frac{k\pi}{4} \mid k = -4, \dots, 3\}$ for training, as introduced in 3.

In this experiment, we evaluate performance across three tasks for each game. Each task involves a policy pretrained with a specific reward function, with the learning agent interacting with this pretrained agent to assess performance. The tasks are:

- *Selfish Agent*: the agent pretrained with its own reward function.
- *Prosocial Agent*: the agent pretrained with the collective reward function.
- *Inequity-Averse Agent*: the agent pretrained with the inequity-averse reward function, as described in (Hughes et al., 2018).

Since the baseline algorithms were not designed for mixed-motive games, we tested them in a cooperative setting by training with collective rewards and evaluating collective rewards in both *Harvest* and *CleanUp* to ensure fairness. Further details on baseline algorithms and experimental settings in these mixed-motive games are provided in B.2.

Table 1: Collective reward of different algorithms in the Harvest and CleanUp.

| Algorithm | Harvest | | | CleanUp | | |
|--------------------------|---------------------|---------------------|---------------------|----------------------|---------------------|---------------------|
| | selfish | prosocial | inequity-averse | selfish | prosocial | inequity-averse |
| AnyPlay | 20.62 (4.91) | 25.13 (5.29) | 3.19 (9.12) | 11.58 (8.41) | 2.42 (12.16) | -1.90 (4.21) |
| Trajedi | 19.74 (5.21) | 23.95 (5.96) | 3.07 (11.06) | -3.41 (20.87) | -23.00 (16.49) | -22.30 (13.05) |
| BRDiv | 12.45 (8.73) | 16.46 (5.57) | -26.97 (17.13) | 8.56 (10.36) | 2.03 (17.47) | -5.38 (9.41) |
| HSP | 21.71 (8.01) | 23.24 (14.23) | 4.95 (10.95) | 10.90 (9.04) | 8.15 (11.86) | -1.20 (3.98) |
| RP ($z=0$) | 6.94 (12.71) | 13.28 (14.06) | 6.35 (5.82) | 13.23 (11.05) | 24.37 (13.45) | 8.55 (11.16) |
| RP ($z=\frac{\pi}{4}$) | 20.46 (3.92) | 23.57 (5.56) | 15.15 (4.35) | 23.35 (13.93) | 38.09 (7.71) | 23.42 (6.65) |
| RP ($z=\frac{\pi}{2}$) | 20.60 (3.52) | 25.84 (3.98) | 15.61 (6.94) | 22.73 (12.58) | 39.53 (6.99) | 18.64 (10.65) |

As shown in Table 1, RP demonstrates comparable performance in both Harvest and CleanUp games, especially in CleanUp game. In Harvest, although AnyPlay and HSP ultimately learn a non-confrontational policy, they contribute little to the collective reward. BRDiv learns a resource-competing policy, leading to resource depletion in the *Selfish Agent* task, and performs moderately in the *Prosocial Agent* task. In the *Inequity-Averse Agent* task, all baselines perform poorly. Instead, RP shows good performance in all tasks, because RP learned to dodge and harvest in interactions with aggressive roles. When RP play an individualistic role ($z = 0$), it tends to use the beam to penalize other agents to get a individual reward, which lowers the collective reward.

In CleanUp, all baselines failed to learn a robust policy that achieves high collective rewards in evaluation tasks, as the environment requires agents to divide the work and cooperate to achieve a high collective outcomes. The policy pools they generated in the first training stage are insufficient to help them learn effective policies. In contrast, RP can utilize the exploration data with cooperative roles to learn a good policy. In the *Selfish Agent* task, RP learns to clean pollution all the time. In the *Prosocial Agent* task, RP learns to clean the pollution while the other agent is harvesting, once the other agent leaves the orchard, RP will harvest apples to maximize collective reward. Since the pretrained inequity-averse agent has lower advantageous inequity aversion³, it converges to harvest apples if they are available in the orchard. Therefore, in the *Inequity-Averse Agent* task, RP learns to clean the pollution continuously. We prefer readers to our project website⁴ for visualizations of the results.

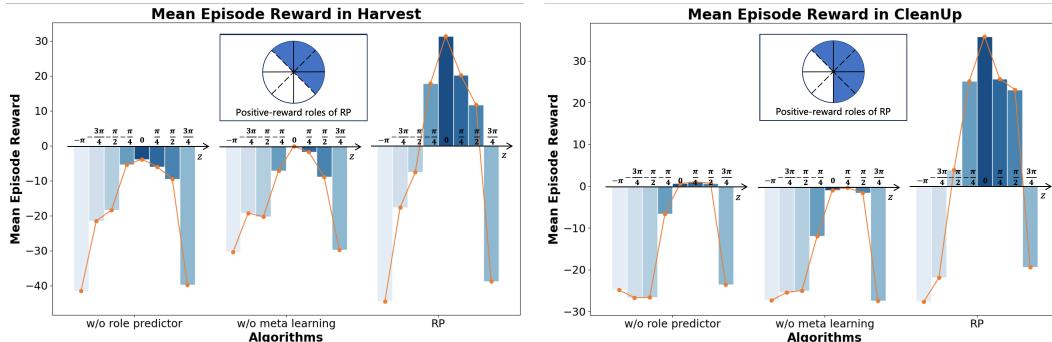


Figure 8: Impact of Role Predictor and Meta-Learning on Performance in Harvest and CleanUp.

5.3 Ablation Studies

In this subsection, we examine the impact of the role predictor and the meta-learning method on the performance of the RP framework through comprehensive ablation studies. We assess the contributions of each component by systematically removing the role predictor and the meta-learning method. We analyze interactions by having each role engage separately with all eight distinct roles. For each unique role pairing, agents interact over a series of 1000 episodes. Given the considerable variation in rewards associated with different roles, we employ the mean episode reward of each role

³Details are shown in B.2

⁴https://weifan408.github.io/role_play_web/

as a metric for assessing performance. The results are shown in Fig. 8. The figure clearly illustrates RP’s superior adaptability to diverse roles compared to configurations lacking the role predictor and meta-learning method. The inset provides a detailed view of RP’s roles that achieve positive average rewards. Roles commonly seen in real-life settingssuch as Altruistic, Prosocial, Individualistic, and Competitiveyield positive rewards. In CleanUp, the Sadistic role also achieves a small positive reward due to its exploitative tendency, though this behavior lacks a virtuous cycle, resulting in only minimal positive gains in average. For the Individualistic role ($z = 0$), RP achieves the highest positive rewards. By contrast, in the ablation configuration without the role predictor and meta-learning method, the rewards are significantly lower even negative. Notably, RP attains positive average rewards when adopting prosocial roles and incurs negative rewards with antisocial roles, whereas policies from the ablation studies consistently produce either negative rewards or considerably lower positive rewards across all roles.

5.4 Role Behavior Analysis

In our analysis of the RP framework, we investigated the principal role behaviors exhibited in the Harvest and CleanUp games. Experimental setting is the same with 5.3. Fig. 9 visualizes our findings for four main roles that reflect real-life behavioral patterns: Competitive ($z = -\frac{\pi}{4}$), Individualistic ($z = 0$), Prosocial ($z = \frac{\pi}{4}$), and Altruistic ($z = \frac{\pi}{2}$).

The Individualistic role, marked by a significantly larger area on the radar chart, suggests an advantageous position in interpersonal dynamics, a phenomenon supported by findings in individualism studies (Triandis, 2018). In contrast, the Competitive role, adept at penalizing others and monopolizing resources such as apples, frequently triggers retaliatory actions from other agents, often culminating in considerable negative rewards. The radar chart shows a large number of blank spaces in the lower half, indicating frequent failures in interactions with antisocial roles.

In the more competitive Harvest game, the Altruistic role demonstrates vulnerabilities as it primarily focuses on the welfare of others, often to its own detriment. Meanwhile, the Prosocial role, which aims to harmonize self-interest with the collective good, shows robust performance in this environments by fostering optimal group outcomes.

Conversely, in the CleanUp game, the Prosocial and Altruistic roles engage in environmentally beneficial behaviors, such as pollution removal. While these actions yield lower direct rewards for themselves, they facilitate greater social welfare for other agents. However, roles driven by self-interest, such as the Competitive and Individualistic roles, often reap benefits from these altruistic actions without bearing any of the associated costs. Additionally, when interacting with the more self-sacrificing Martyr role, both the Prosocial and Altruistic roles experience enhanced social welfare.

Overall, the role behavior analysis demonstrates the adaptability of the RP framework across diverse roles, with each role maintaining its distinct SVO.

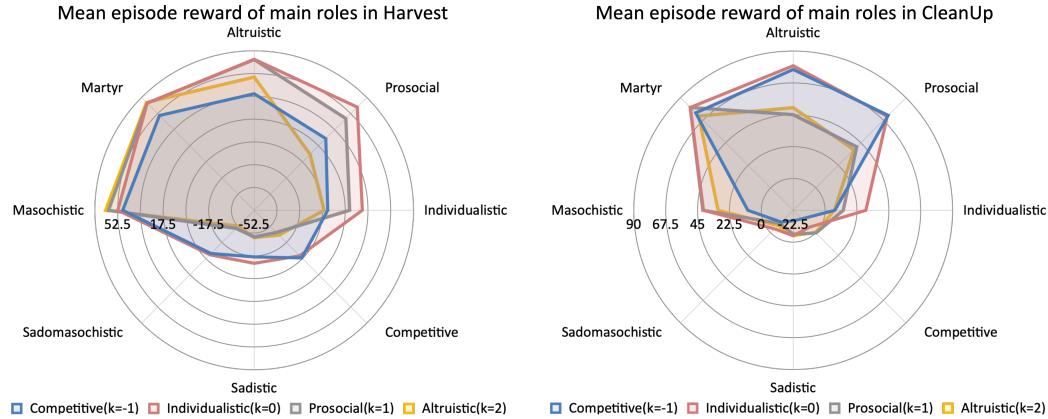


Figure 9: Role analysis in Harvest and CleanUp.

5.5 Prediction Analysis

In the RP framework, the role predictor is utilized to estimate the joint role embeddings of other agents. Fig. 10 displays the prediction outcomes of the role predictor $q(\phi)$ within both the Harvest and CleanUp games. Each heatmap represents the prediction results of a role interacting with all roles. The colors in the heatmap indicate the proportion of predictions that align with actual roles, with colors closer to the diagonal representing higher accuracy. This visualization method clearly illustrates that the most accurate predictions are those where the color intensity peaks along the diagonal, indicating a strong alignment between predicted and actual roles. These results confirm the efficacy of the role predictor in accurately forecasting the role embeddings of other agents. Notably, the predicted roles align with the actual roles or their close counterparts. Although prediction performance is slightly reduced in the CleanUp game due to its more complex environmental dynamics compared to the Harvest game, the role predictor occasionally outputs incorrect roles that exhibit similar behaviors to the actual roles, which still aids the learning agent in adapting to the environment.

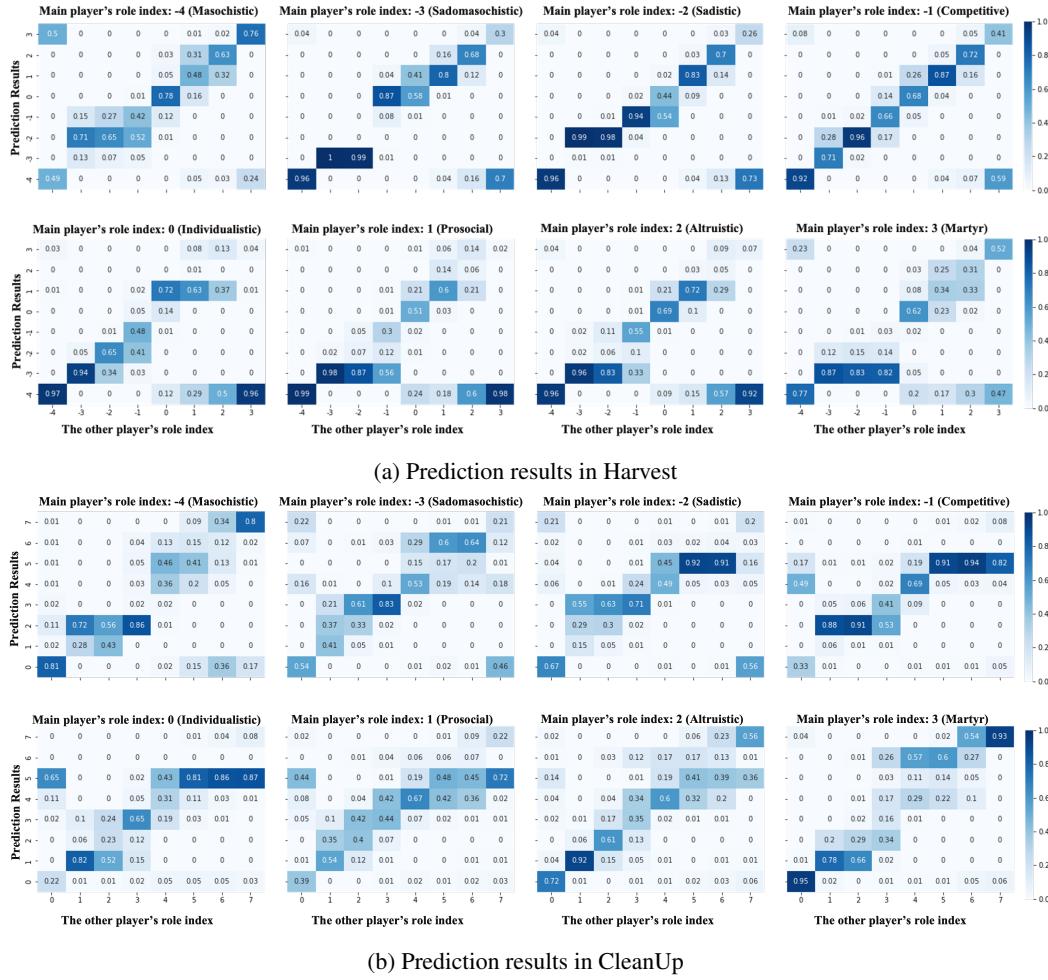


Figure 10: Prediction results of the role predictor $q(\phi)$.

6 Conclusion

In this study, we introduced a novel framework called Role Play (RP) to enable agents to adapt varying roles, effectively addressing the zero-shot coordination challenge in MARL. Drawing inspiration from social intuition, RP incorporates a role predictor that estimates the joint role embeddings of other agents, which in turn facilitates the adaptation of the learning agent's role based on these pre-

dictions. We demonstrated the effectiveness of RP across both cooperative and mixed-motive environments, where agents must balance individual contributions against collective benefits, achieving comparable performance compared to state-of-the-art baselines.

However, the RP framework is not without its limitations. As the number of agents increases, the complexity of role prediction escalates, potentially challenging the ability of role predictor to accurately forecast the joint role embeddings of other agents. Additionally, the reward feature mapping function ψ plays a crucial role in the framework, with its selection significantly influencing the learning process.

Future work will aim to tackle these challenges by enhancing the accuracy of role prediction and optimizing the choice of the reward feature mapping function. We also plan to extend the RP framework to more complex multi-agent scenarios, further testing its scalability and adaptability.

References

- Bai, Y., Jin, C., and Yu, T. (2020). Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170.
- Beck, J., Vuorio, R., Liu, E. Z., Xiong, Z., Zintgraf, L., Finn, C., and Whiteson, S. (2023). A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*.
- Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.
- Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., and Dragan, A. (2019). On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32.
- Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. (2016). RL2: Fast reinforcement learning via slow reinforcement learning. 2016. URL <http://arxiv.org/abs/1611.02779>.
- Fakoor, R., Chaudhari, P., Soatto, S., and Smola, A. J. (2019). Meta-q-learning. *arXiv preprint arXiv:1910.00125*.
- Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.
- Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. (2017). Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*.
- Garnelo, M., Czarnecki, W. M., Liu, S., Tirumala, D., Oh, J., Gidel, G., van Hasselt, H., and Balduzzi, D. (2021). Pick your battles: Interaction graphs as population-level objectives for strategic diversity. *arXiv preprint arXiv:2110.04041*.
- Gibbons, R. et al. (1992). A primer in game theory.
- Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. (2020). other-play for zero-shot coordination. In *International Conference on Machine Learning*, pages 4399–4410. PMLR.
- Hughes, E., Leibo, J. Z., Phillips, M., Tuyls, K., Dueñez-Guzman, E., García Castañeda, A., Dunning, I., Zhu, T., McKee, K., Koster, R., et al. (2018). Inequity aversion improves cooperation in intertemporal social dilemmas. *Advances in neural information processing systems*, 31.
- Jellema, T., Macinska, S. T., OConnor, R. J., and Skodova, T. (2024). Social intuition: behavioral and neurobiological considerations. *Frontiers in Psychology*, 15:1336363.
- Kim, D. K., Liu, M., Riemer, M. D., Sun, C., Abdulhai, M., Habibi, G., Lopez-Cot, S., Tesauro, G., and How, J. (2021). A policy gradient algorithm for learning to learn in multiagent reinforcement learning. In *International Conference on Machine Learning*, pages 5541–5550. PMLR.
- Kirk, R., Zhang, A., Grefenstette, E., and Rocktäschel, T. (2021). A survey of generalisation in deep reinforcement learning. *arXiv preprint arXiv:2111.09794*, 1:16.
- Ko, S. (2006). Mathematical analysis.

- Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., Silver, D., and Graepel, T. (2017). A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30.
- Leibo, J. Z., nez Guzmán, E. D., Vezhnevets, A. S., Agapiou, J. P., Sunehag, P., Koster, R., Matyas, J., Beattie, C., Mordatch, I., and Graepel, T. (2021). Scalable evaluation of multi-agent reinforcement learning with melting pot. PMLR.
- Lieberman, M. D. (2000). Intuition: a social cognitive neuroscience approach. *Psychological bulletin*, 126(1):109.
- Lu, C., Willi, T., De Witt, C. A. S., and Foerster, J. (2022). Model-free opponent shaping. In *International Conference on Machine Learning*, pages 14398–14411. PMLR.
- Lucas, K. and Allen, R. E. (2022). Any-play: An intrinsic augmentation for zero-shot coordination. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’22, pages 853–861, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- Lupu, A., Cui, B., Hu, H., and Foerster, J. (2021). Trajectory diversity for zero-shot coordination. In *International conference on machine learning*, pages 7204–7213. PMLR.
- McKee, K. R., Gemp, I., McWilliams, B., Duèñez Guzmán, E. A., Hughes, E., and Leibo, J. Z. (2020). Social diversity and social preferences in mixed-motive reinforcement learning. AAMAS ’20, pages 869–877, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- Mitchell, E., Rafailov, R., Peng, X. B., Levine, S., and Finn, C. (2021). Offline meta-reinforcement learning with advantage weighting. In *International Conference on Machine Learning*, pages 7780–7791. PMLR.
- Murphy, R. O. and Ackermann, K. A. (2014). Social value orientation: Theoretical and measurement issues in the study of social preferences. *Personality and Social Psychology Review*, 18(1):13–41.
- Peng, Z., Li, Q., Hui, K. M., Liu, C., and Zhou, B. (2021). Learning to simulate self-driven particles system with coordinated policy optimization. *Advances in Neural Information Processing Systems*, 34:10784–10797.
- Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., and Botvinick, M. (2018). Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR.
- Rahman, A., Fosong, E., Carlucho, I., and Albrecht, S. V. (2023). Generating teammates for training robust ad hoc teamwork agents via best-response diversity. *Transactions on Machine Learning Research*.
- Raileanu, R., Denton, E., Szlam, A., and Fergus, R. (2018). Modeling others using oneself in multi-agent reinforcement learning. In *International conference on machine learning*, pages 4257–4266. PMLR.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Schwarting, W., Pierson, A., Alonso-Mora, J., Karaman, S., and Rus, D. (2019). Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50):24972–24978.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.
- Song, X., Gao, W., Yang, Y., Choromanski, K., Pacchiano, A., and Tang, Y. (2019). Es-maml: Simple hessian-free meta learning. *arXiv preprint arXiv:1910.01215*.
- Strouse, D., McKee, K., Botvinick, M., Hughes, E., and Everett, R. (2021). Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34:14502–14515.
- Tack, J., Park, J., Lee, H., Lee, J., and Shin, J. (2022). Meta-learning with self-improving momentum target. *Advances in Neural Information Processing Systems*, 35:6318–6332.
- Triandis, H. C. (2018). *Individualism and collectivism*. Routledge.
- Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.
- Yu, C., Gao, J., Liu, W., Xu, B., Tang, H., Yang, J., Wang, Y., and Wu, Y. (2023). Learning zero-shot cooperation with humans, assuming humans are biased. *arXiv preprint arXiv:2302.01605*.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. (2020). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR.

Zhao, R., Song, J., Yuan, Y., Hu, H., Gao, Y., Wu, Y., Sun, Z., and Yang, W. (2023). Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6145–6153.

A Theorem Proof

Theorem 4.1. *For a finite MDP with T time steps and a specific role policy $\pi(z)$, if any random policy π' is ϵ -close to the role policy $\pi(z')$, then we have*

$$\left| \frac{J(\pi(z), \pi')}{J(\pi(z), \pi(z'))} - 1 \right| \leq \epsilon T. \quad (10)$$

Proof. We start by examining the trajectory probabilities under policies $\pi(z)$ and π' :

$$p_{\pi(z), \pi'}(\tau) = p(s_0) \prod_t^T \pi(a_t | o_t, z) \pi'(a'_t | o'_t) p(s_{t+1} | s_t, a_t, a'_t). \quad (11)$$

Given the assumption

$$\left| \frac{\pi'(a'_t | o'_t)}{\pi(a'_t | o'_t, z)} - 1 \right| < \epsilon, \quad (12)$$

we can approximate the ratio of trajectory probabilities under π' and $\pi(z')$:

$$\left| \frac{p_{\pi(z), \pi'}(\tau)}{p_{\pi(z), \pi(z')}(\tau)} - 1 \right| \approx \left| \prod_{t=0}^T \frac{\pi'(a'_t | o'_t)}{\pi(a'_t | o'_t, z)} - 1 \right| \approx \left| \left(\sum_{t=0}^T \frac{\pi'(a'_t | o'_t)}{\pi(a'_t | o'_t, z)} - 1 \right) \right| \leq \epsilon T. \quad (13)$$

And the expected reward can then be written as:

$$J(\pi(z), \pi') = \sum_{\tau} p_{\pi(z), \pi'}(\tau) \psi(\mathcal{R}(\tau), z), \quad (14)$$

where

$$\mathcal{R}(\tau) = \sum_t^T \mathcal{R}(s_t, a_t, a'_t), \quad (15)$$

and ψ is a predefined mapping function. Therefore, we have

$$\left| \frac{J(\pi(z), \pi')}{J(\pi(z), \pi(z'))} - 1 \right| \leq \epsilon T. \quad (16)$$

□

B Experimental Details

The baselines were established using the implementation details provided in the repositories from Rahman et al. (2023) (AnyPlay, TrajeDi, BRDiv)⁵ and Yu et al. (2023) (HSP)⁶. The network architecture for these baselines is identical to that of the RP framework, which is detailed below. The additional role predictor q_ϕ for RP is a one-layer fully connected neural network with 64 hidden units and ReLU activation function. We used the OpenAI gym implementation of Overcooked in HSP (Yu et al., 2023) and adapted it for other algorithms. The Harvest and CleanUp games were implemented using the MeltingPot framework (Leibo et al., 2021). Detailed experimental settings for Overcooked, Harvest, and CleanUp are provided in the following subsections. For more details, please refer to our code repository.⁷

⁵<https://github.com/ueo-agents/BRDiv>

⁶<https://github.com/samjia2000/HSP>

⁷https://github.com/Weifan408/role_play

B.1 Experimental Details of Overcooked

The reward shaping function in Overcooked is defined as:

$$\psi(r^i, z^i) = r^i + \sum_k (z_k^i * E_k), \quad (17)$$

where $z_k^i \in [-1, 0, 1]$ is the preference of agent i for event k , and E_k is the reward associated with event k . The reward shaping function is designed to encourage agents to exhibit specific behaviors based on their role preferences. Table 2 shows the event reward we set in Overcooked. HSP uses the same reward shaping function as RP, while AnyPlay, TrajeDi, and BRDiv use the original collective reward directly from the environment.

Table 2: Event reward in Overcooked.

| Event | Reward |
|---|--------|
| Picking up an item from any dispenser (onion, tomato, dish) | 5 |
| Picking up a soup | 5 |
| Viable placement | 5 |
| Optimal placement | 5 |
| Catastrophic placement | 10 |
| Placing an item into the pot (onion, tomato) | 3 |
| Delivery | 10 |

Table 3 shows the detailed model architecture used in Overcooked. The policy and value networks are trained using the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017).

Table 3: Base model architecture used in Overcooked.

| Layer | Architecture |
|-----------------------|------------------------------|
| CNN Filter | [[32,3,1],[64,3,1],[32,3,1]] |
| CNN Activation | ReLU |
| LSTM Cell Size | 128 |
| LSTM Activation | SiLU |
| Post Fcnet Hiddens | [64,64] |
| Post Fcnet Activation | ReLU |

B.2 Experimental Details in Harvest and CleanUp

We employ ring formulation of SVO to categorize agent roles. We introduced SVO-based (role-based) rewards as intrinsic motivators for agents to learn role-specific strategies. The reward feature mapping function ψ is defined as:

$$\psi(\mathbf{r}, z^i) = w * r^i + (1 - w) * \underbrace{\left[\cos(z^i)r^i + \sin(z^i)\bar{r}^{-i} \right]}_{\text{SVO-based reward shaping}}, \quad (18)$$

where r^i denotes the individual reward for agent i , \bar{r}^{-i} the average reward of other agents, and w a hyperparameter adjusting the impact of role-based rewards. In implementation, we set $w = 0.3$, and the role embedding z^i is one-hot encoded. Table 4 shows the hyperparameters used in Harvest and CleanUp.

We pre-trained three distinct policies using a naive SP framework and PPO for zero-shot evaluation: a selfish agent, a prosocial agent, and an inequity-averse agent (Hughes et al., 2018). Each agent is trained with a different reward function tailored to its role:

- **Selfish Agent:** This agent is trained with the original reward directly from the environment.
- **Prosocial Agent:** This agent is trained using a collective reward that sums the rewards of all agents, formalized as $r_i = \sum_j r_j$.

Table 4: Hyperparameters in Harvest and CleanUp.

| Hyperparameters | Values |
|------------------|-----------|
| FCnet Hiddens | [256,256] |
| FCnet Activation | Tanh |
| LSTM Cell Size | 256 |
| LSTM Activation | SiLU |
| w | 0.3 |
| Trail Length | 10 |

- **Inequity-Averse Agent:** This agent’s training includes an inequity shaping reward designed to address social fairness, defined by the formula:

$$r_i = r_i - \frac{\alpha \sum_{j \neq i} \max(0, r_j - r_i) + \beta \sum_{j \neq i} \max(0, r_i - r_j)}{n - 1},$$

where $\alpha \sum_{j \neq i} \max(0, r_j - r_i)$ and $\beta \sum_{j \neq i} \max(0, r_i - r_j)$ are the penalty for disadvantageous inequity and the reward for advantageous inequity, respectively. For this agent, $\alpha = 5$ and $\beta = 0.05$ are used, following the default values set in (Hughes et al., 2018).

Additionally, we modified the *attack* action for both the selfish and inequity-averse agents to *stay* in the CleanUp game to prevent them from becoming non-responsive (*wooden man*) after training.