# Learning Roles with Emergent Social Value Orientations

Wenhao Li[*] Xiangfeng Wang[†] Bo Jin[‡] Jingyi Lu[§] Hongyuan Zha[¶]

February 1, 2023

**Abstract**

Social dilemmas can be considered situations where individual rationality leads to collective irrationality. The multi-agent reinforcement learning community has leveraged ideas from social science, such as social value orientations (SVO), to solve social dilemmas in complex cooperative tasks. In this paper, by first introducing the typical "division of labor or roles" mechanism in human society, we provide a promising solution for intertemporal social dilemmas (ISD) with SVOs. A novel learning framework, called Learning **R**oles with **E**mergent **SVO**s (**RESVO**), is proposed to transform the learning of roles into the social value orientation emergence, which is symmetrically solved by endowing agents with altruism to share rewards with other agents. An SVO-based role embedding space is then constructed by individual conditioning policies on roles with a novel rank regularizer and mutual information maximizer. Experiments show that RESVO achieves a stable division of labor and cooperation in ISDs with different complexity.

## 1 Introduction

The continuity of human civilization and the prosperity of the race depends on our ability to cooperate. From evolutionary biology to social psychology and economics, cooperation in human populations has been regarded as a paradox and a challenge (Fehr and Fischbacher, 2003; Pennisi, 2009; Santos et al., 2021). Cooperation issues vary in scale and are widespread in daily human life, ranging from assembly line operations in factories and scheduling of seminars to peace summits between significant powers, business development, and pandemic control (Dafoe et al., 2020).

Although cooperation can benefit all parties, it might be costly. Thus, the temptation to evade any cost (i.e., the free-riding) becomes a tempting strategy, which leads to cooperation collapsing, or the multi-person social dilemma (Rapoport et al., 1965; Xu et al., 2019). That is, "individually reasonable behavior leads to a situation in which everyone is worse off than they might have been otherwise" (Kollock, 1998). Just as cooperation widely exists in human social, economic, and political activities, most thorny problems we face, from the interpersonal to the international, are at their core social dilemmas. This article presents two cases in recent years closely related to the future economic and political decisions of countries, namely autonomous driving and carbon trading, and the role of social dilemmas in them.

Autonomous driving (AV), which promises world-changing benefits by increasing traffic efficiency (Van Arem et al., 2006), reducing pollution (Spieser et al., 2014), and eliminating up to 90% of traffic accidents (Gao et al., 2014), is a very complex systems engineering. Existing work mainly focuses on accomplishing generic tasks, such as following a planned path while obeying traffic rules. However, there are many driving scenarios in practice, most of which have social dilemmas. Examples include lane changing (Dafoe et al., 2020), meeting, parking (Li, 2022), and even ethical aspects of aggressive versus conservative driving behavior

[*] The Chinese University of Hong Kong, Shenzhen; liwenhao@cuhk.edu.cn

[†] East China Normal University; xfwang@cs.ecnu.edu.cn

[‡] East China Normal University; bjin@cs.ecnu.edu.cn

[§] East China Normal University; jylu@psy.ecnu.edu.cn

[¶] The Chinese University of Hong Kong, Shenzhen; zhahy@cuhk.edu.cn

choices (Bonnefon et al., 2016). Therefore, the practicality of AV depends on the efficient solution to social dilemmas.

Carbon trading is a greenhouse gas emission right (emission reduction) transaction based on the United Nations Framework Convention on Climate Change established by the Kyoto Protocol to promote the reduction of greenhouse gas emissions, using a market mechanism (Grimeaud, 2001). Carbon emission is a representative social dilemma in which countries' direct gas emissions for the sake of economic development undermine collective interests. The typical mechanisms in carbon trading, such as *distribution of allowances* (Fullerton and Metcalf, 2014), *joint implementation* (Grimeaud, 2001), etc., have obvious correspondences with the *boundaries* (Ibrahim et al., 2020a,b) and *institutions* (Koster et al., 2020; Lupu and Precup, 2020) used to solve social dilemmas in economics and social psychology.

The social dilemma has been comprehensively studied in economics, social psychology, and evolutionary biology in the past few decades. This paper focuses on the *public good dilemma* in the intertemporal social dilemma (ISD). A public good is a resource from which all may benefit, regardless of whether they have helped provide the good (producer) (Kollock, 1998). This is to say that public goods are *non-excludable*. As a result, there is the temptation to enjoy the good (consumer) without contributing to its creation or maintenance. Those who do so are termed *free-riders*, and while it is individually rational to free-ride if all do so, the public good is not provided, and all are worse off.

Artificial intelligence (AI) advances pose increasing opportunities for AI research to promote human cooperation and enable new tools for facilitating cooperation (Dafoe et al., 2020). Recently, multi-agent reinforcement learning (MARL) has been utilized as a powerful toolset to study human cooperative behavior with great success (Lowe et al., 2017; Silver et al., 2018; Jaderberg et al., 2019; Liao et al., 2020; Li et al., 2022). We believe it is reasonable to use MARL as a first step in exploring the use of AI tools to study multi-person social dilemmas. The current model for reinforcement learning suggests that reward maximization is sufficient to drive behavior that exhibits abilities studied in the human cooperation and social dilemmas, including "knowledge, learning, perception, social intelligence, language, generalization and imitation" (Yang, 2021; Silver et al., 2021; Vamplew et al., 2022). The justification for this claim is deeply rooted in the *von Neumann Morgenstern utility theory* (von Neumann and Morgenstern, 2007), which is the basis for the well-known *expected utility theory* (Schoemaker, 2013) and essentially states that it is safe to assume an intelligent entity will always make decisions according to the highest expected utility in any complex scenarios[1] (Yang, 2021).

In MARL, the critical issue of multi-person social dilemma can be formalized as an ISD (Leibo et al., 2017; Hughes et al., 2018b), and most MARL methods have introduced ideas from social psychology and economics more or less. These methods could be divided into three categories, *strategic* solutions, *structural* solutions, and *motivational* solutions, based on whether the solutions assume egoistic agents and whether the structure of the game can be changed (Kollock, 1998) according to the taxonomy of social science.

*Structural* solutions reduce the difficulty of the original social dilemma by changing the game's rules or completely avoiding the occurrence of the social dilemma. The mechanisms introduced into MARL mainly include boundaries and sanctions (Ostrom, 1990). Ibrahim et al. (2020a) indirectly sets boundaries for resources by introducing a shared periodic signal and a conditional policy based on this signal, allowing agents to access shared resources in a fixed order. Ibrahim et al. (2020b) achieves resource boundarization by introducing a centralized government module through taxation and wealth redistribution.. Koster et al. (2020); Lupu and Precup (2020) introduce a centralized module and use rules and learning methods to punish the free-riding agent separately. LIO (Yang et al., 2020) enables each agent to punish, thereby implementing the sanction mechanism in a decentralized manner. Vinitsky et al. (2021) adopts a combination of centralized and decentralized modules and judges the decentralized sanctioning behavior of the agent through the centralized module, thereby encouraging appropriate sanctioning behaviors and avoiding unreasonable behaviors. Furthermore, Dong et al. (2021) introduces homophily into the MARL to solve the second-order social dilemma caused by sanctions.

*Strategic* solutions assume that all individuals in the group are egoists and that the algorithm does

[1] Although follow-up works have shown that some of the assumptions on rationality could be violated by real decision-makers in practice (Gigerenzer and Selten, 2002), those conditions are rather taken as the "axioms" of rational decision making (Yang, 2021).

not change the game's structure. Such methods rely on an individual's ability to shape other individuals' payoffs, thereby directly influencing the behavior of others. Direct and indirect reciprocity is the main mechanisms introduced into MARL. Eccles et al. (2019) introduces the classic direct reciprocity algorithm tit-for-tat (Axelrod and Hamilton, 1981) into the solution of ISD. In order to realize the "imitation" at the core of tit-for-tat and the definition of the binary action (cooperate and defect) in ISD, Eccles et al. (2019) divides the agents into innovators and imitators and introduces the niceness function based on the deep advantage function. Anastassacos et al. (2021) introduces two core concepts of indirect reciprocity, reputation and social norm (Santos et al., 2021) into MARL and uses them as fixed rules to construct the agent's action space.

*Motivational* solutions assume agents are not entirely egoistic and so give some attention (passively or actively) to the outcomes of their partners. One of the typical mechanisms is communication. Across a wide variety of economics and social psychology studies, when individuals are given a chance to talk with each other, cooperation increases significantly (Orbell et al., 1988, 1990). Although there are many works (Sheng et al., 2020; Ahilan and Dayan, 2021) on communication learning in MARL, little attention has been paid to the role of communication in solving ISD. Pretorius et al. (2020) first uses empirical game-theoretic analysis (Tuyls et al., 2018) to study existing communication learning methods in ISD and to verify the effects of these methods experimentally. Another typical mechanism is social value orientation. Social value orientations (SVOs), or heterogeneous distributive preferences (Batson, 2012; Cooper and Kagel, 2016; Eckel and Grossman, 1996; Rushton et al., 1981; Simon, 1993), are widely recognized in social psychology and economics as an effective mechanism for promoting the emergence of human cooperative behavior in different social dilemmas (McKee et al., 2020).

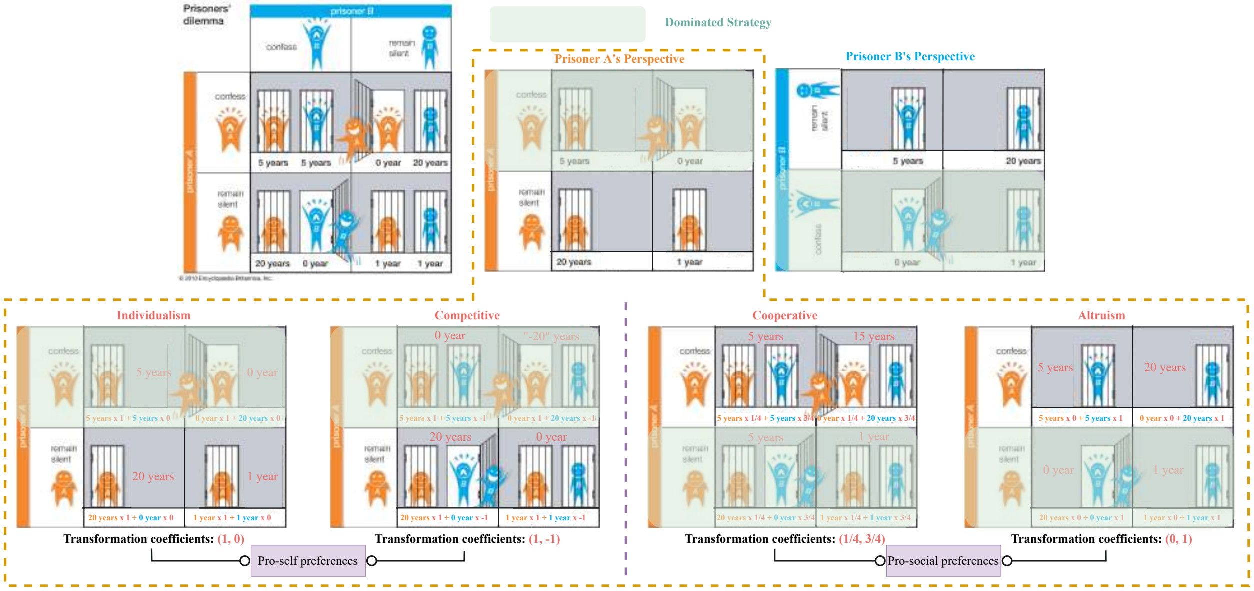

Prisoners' dilemma
prisoner B
confess
remain silent
prisoner A
5 years
0 year
20 years
1 year
© 2010 Encyclopædia Britannica, Inc.
Dominated Strategy
Prisoner A's Perspective
Prisoner B's Perspective
Individualism
Competitive
Cooperative
Altruism
5 years x 1 + 5 years x 0
0 year x 1 + 20 years x 0
20 years x 1 + 0 year x 0
1 year x 1 + 1 year x 0
"-20" years
15 years
5 years x 1 + 5 years x -1
0 year x 1 + 20 years x -1
20 years x 1 + 0 year x -1
1 year x 1 + 1 year x -1
5 years x 1/4 + 5 years x 3/4
0 year x 1/4 + 20 years x 3/4
20 years x 1/4 + 0 year x 3/4
1 year x 1/4 + 1 year x 3/4
5 years x 0 + 5 years x 1
0 year x 0 + 20 years x 1
20 years x 0 + 0 year x 1
1 year x 0 + 1 year x 1
Transformation coefficients: (1, 0)
Transformation coefficients: (1, -1)
Transformation coefficients: (1/4, 3/4)
Transformation coefficients: (0, 1)
Pro-self preferences
Pro-social preferences

Figure 1: Interdependence theory in the prisoner's dilemma (Encyclopædia Britannica, 2022): the four pathways depict transformation processes for a row player who has individualistic, competitive, cooperative, and altruistic preferences, respectively; four resulting transformations suggest different dominant strategies (highlighted in green).

The above three types of methods mainly make breakthroughs in methodology and are accompanied by simulation experiments to verify the correctness of the conclusions. Considering the completeness of the theory and the feasibility of convergence analysis, this paper mainly focuses on solving intertemporal or public good social dilemmas based on social value orientations. The aforementioned mainstream conclusions about SVO from social psychology and economics are mainly supported by interdependence theory (Hansen, 1982). In social psychology and economics games, classical game theory does not accurately predict human behavior. This is because, in these human-involved games, each player does not rely on the given payoff matrix to make decisions but on their own "effective" payoff matrix (Hansen, 1982; McKee et al., 2020). The effective payoff matrix is constructed by redistributing payoffs for the given payoff matrix based on the players' respective SVOs. As seen from Figure 1, different SVO will make players choose different dominant strategies when facing the prisoner's dilemma, thus affecting the emergence of cooperation. Many different social value orientations are theoretically possible, but most work has concentrated on various linear combinations of individuals' concern for the rewards for themselves and their partners.

Inspired by the interdependence theory, many previous works have introduced the SVO into MARL to solve the ISD (Peysakhovich and Lerer, 2018b; Hughes et al., 2018a; Zhang et al., 2019; Wang et al., 2019; Baker, 2020; Gemp et al., 2022; Yi et al., 2021; Ivanov et al., 2021; Schmid et al., 2021). Peysakhovich and Lerer (2018b) introduces the SVO into MARL for the first time and proposes the concept of prosocial, that is, cooperative orientation agents. The reward function of a prosocial agent is shaped as a fixed linear combination of its reward and the others. Hughes et al. (2018a) introduces an inequity aversion model in ISD, namely equality orientation, which promotes cooperation by minimizing the gap between one's return and that of other individuals. The latter work is no longer satisfied with a fixed linear combination and begins to introduce trainable weight parameters. Baker (2020) first attempts to randomize the linear weights of one's and others' rewards to observe whether cooperative behavior emerges. Since the linear weights are always greater than 0, all agents can be roughly classified into three categories: cooperative-oriented, altruistic-oriented, or individual-oriented. Going a step further, D3C (Gemp et al., 2022) optimizes the linear combination weights by using the ratio of the worst equilibrium to the optimal solution (Price of Anarchy, PoA) that measures the quality of the equilibrium points. Concurrent work LToS (Yi et al., 2021) models the optimization problem of linearly transforming weights as a bi-level problem and uses an end-to-end approach to train weights and policies jointly. Considering the noise or privacy issues that instantaneous rewards for SVO modeling may introduce, some recent works shape the agents' reward in other ways. Schmid et al. (2021) realizes the conditional linear combination of agent rewards by introducing the idea of the market economy. Zhang et al. (2019) and Ivanov et al. (2021) use state-value and action-value functions to implement SVO modeling. Wang et al. (2019) directly uses reward-to-go and reward-to-come, combined with evolutionary algorithms, to optimize the weights of nonlinear (MLP-based) combinations. However, these methods cannot stably and efficiently converge to mutual cooperation under complex ISDs, which are further verified in our numerical experiments in Section 4.

The conceptual diagram of our solution is shown in Figure 2. Specifically, we find that a typical mechanism of human society, i.e., division of labor or roles, can benefit from providing a promising solution for the ISD combined with SVOs. The effectiveness of the division of labor in solving the ISD has emerged in existing MARL works but is still underexplored. The numerical results from sanction-based methods Yang et al. (2020); Vinitsky et al. (2021) on the typical ISD task *Cleanup* (Hughes et al., 2018b) and *Allelopathic Harvest* (Köster et al., 2020) show that policies solving ISDs effectively exhibit a clear division of labor (Figure 3). Many natural systems feature emergent division of labor, such as ants (Gordon, 1996), bees (Jeanson et al., 2005), and humans (Butler, 2012). In these systems, the division of labor is closely related to the roles and is critical to labor efficiency. The division of labor, or the role theory, has been widely studied in sociology and economics (Institute, 2013). A role is a comprehensive pattern of behavior, and agents with different roles will show different behaviors. Thus the overall performance can be improved by learning from others' strengths (Wang et al., 2020). These benefits inspired multi-agent system designers, who try to reduce the design complexity by decomposing the task and specializing agents with the same role to certain sub-tasks (Wooldridge et al., 2000; Omicini, 2000; Padgham and Winikoff, 2002; Pavón and Gómez-Sanz, 2003; Cossentino et al., 2005; Zhu and Zhou, 2008; Spanoudakis and Moraitis, 2010; DeLoach and Garcia-Ojeda, 2010; Bonjean et al., 2014). However, roles and the associated responsibilities (or subtask-specific rewards Sun et al. (2020)) are predefined using prior knowledge in this systems (Lhaksmana et al., 2018). Although pre-definition can be efficient in tasks with a clear structure, such as software engineering (Bresciani et al., 2004), it hurts generalization and requires prior knowledge that may not be available in practice. To solve this problem, Wilson et al. (2010) uses Bayesian inference to learn a set of roles, and ROMA (Wang

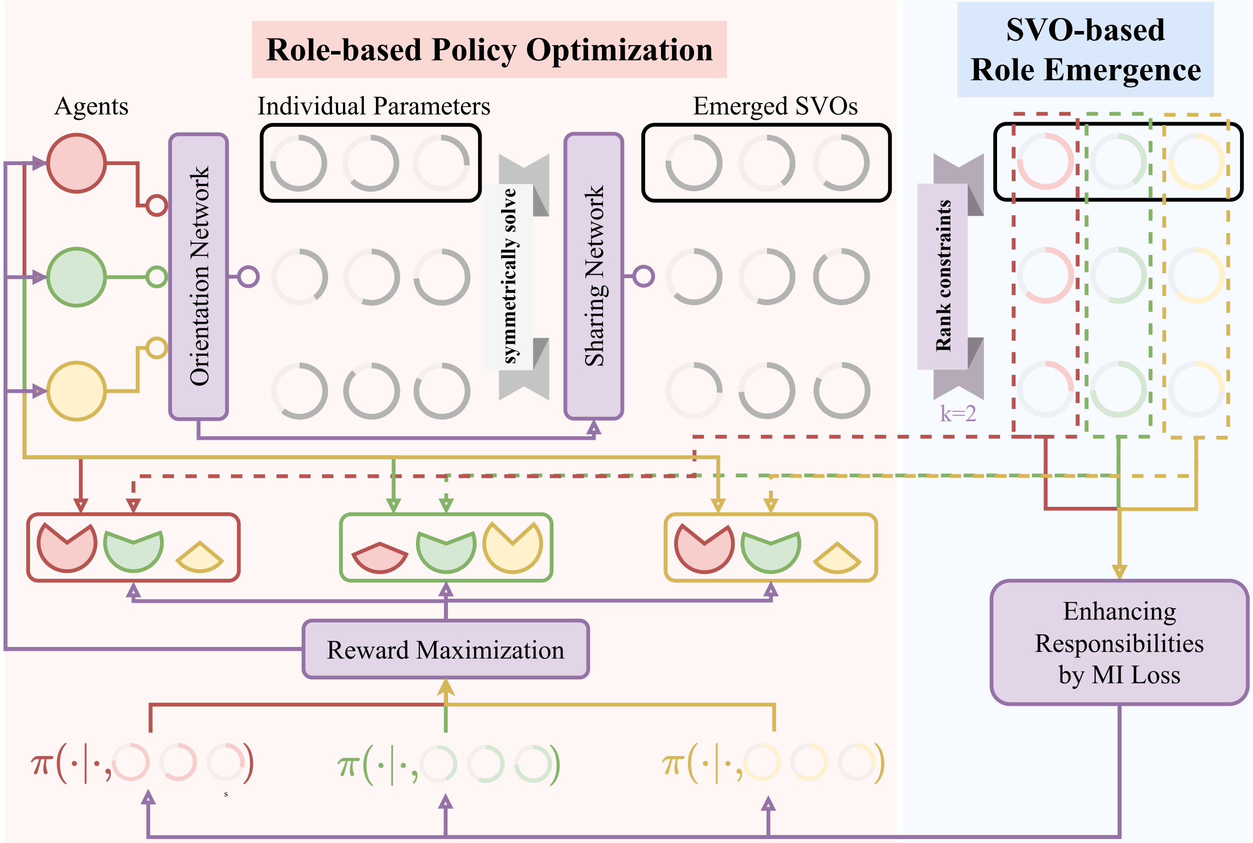
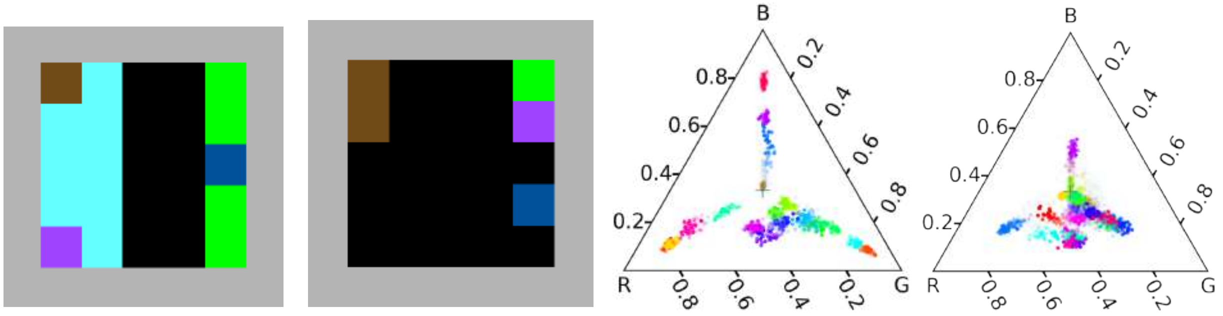


Figure 2: The conceptual diagram of the proposed RESVO, which is based on the social value orientations combined with a typical "division of labor or roles" mechanism, can benefit from providing a promising solution for the intertemporal social dilemma. RESVO is divided into two training phases of joint optimization and interleaved update: SVO-based role or division of labor emergence and role or division-based policy optimization. In the first phase, RESVO transforms the learning of roles into a social value orientation emergence problem, which is symmetrically solved by endowing agents with altruism to learn to share rewards with other agents. An SVO-based role embedding space is then constructed by conditioning individual policies on roles with a novel rank regularizer and mutual information maximizer. Moreover, RESVO optimizes the policies based on the multi-agent policy gradient theorem in the second phase by maximizing the shaped rewards of all agents with different emerged social value orientations.

et al., 2020) designs a specialization objective to encourage the emergence of roles. Wang et al. (2021) improves the learning efficiency in hard-exploration tasks by first decomposing joint action spaces according to action effects, which makes role discovery much more effortless. Unfortunately, none of these methods considers the intertemporal social dilemma.

Drawing the insight from studies in social psychology that characteristics of laborers, or roles, influence the SVOs reciprocally (Sutin and Costa, 2010; Holman and Hughes, 2021), this paper uses the agent's SVO to represent the role of each agent, transforming the role learning problem into the emergence of the agent's SVO, thereby naturally constructing a role-based framework in MARL to solve ISD. Specifically, we use the SVOs, i.e., the coefficients in the transformation matrices from independence theory, to represent each agent's role, e.g., $(0, 1)$ in individualistic preference and $(0.5, 0.5)$ in cooperative preference. This method assumes that all agents will have real-time access to one another's rewards while learning. However, making reward data unrestrictedly accessible is undesirable for several reasons. For example, agent designers



(a) (b) (c) (d)

Figure 3: (a) is a snapshot of the division of labor found by Yang et al. (2020) in *Cleanup* task, where the blue agent picks apples, and the purple one stays on the riverside to clean waste. In contrast, (b) shows a jointly suboptimal division where two failure agents compete for apples. (c-d) Vinitsky et al. (2021) shows similar results in *Allelopathic Harvest* task.

want to imperceptibly modify the agent's reward function or prohibit from sharing their agents' reward function (Kairouz et al., 2021). This makes the emergence of social value orientation unfeasible, making it impossible to promote the division of labor based on SVOs.

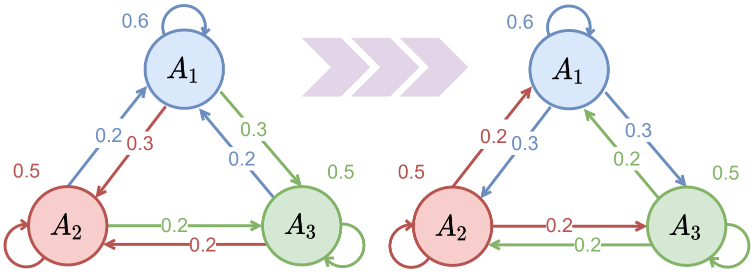


Figure 4: Symmetrically converting (left) the social value orientation learning problem to (right) the learning to share problem in a three-agents environment. The circles of different colors represent different agents, and the numbers of different colors represent the parameters that each agent needs to learn. Assuming that $r_1$, $r_2$, and $r_3$ represent the extrinsic reward of each agent, the shaped reward of $A_1$ is $0.6r_1 + 0.2r_2 + 0.2r_3$. The shaped rewards of other agents can also be computed similarly.

Inspired by the fact that *altruism* plays a crucial role in human's solution to social dilemmas (Kollock, 1998; Eisenberg and Mussen, 1989), that is, consumers altruistic share a part of their profits with producers, this paper proposes a novel algorithm framework, called Learning Role with Emergent SVO (RESVO), that establish a symmetric relationship between SVO emergence and learning to share. RESVO encourages agents to learn to dynamically share the reward with other agents (see Figure 4). In this learning paradigm, the learnable parameters, or the SVO of each agent, are the proportions[2] of rewards it receives from other agents to the extrinsic rewards of these reward giver. We take these emergent SVOs as the role representations of each agent. RESVO then imposes a novel low-rank constraint on the SVO matrix of all agents to effectively represent the different roles of agents and uses projected gradient descent to solve constrained optimization problems. Furthermore, to establish the connection between roles and decentralized policies, RESVO conditions agents' policies on individual emergent SVOs by explicitly feeding agents' SVO-based role embeddings into their local policies correspondingly. Furthermore, to associate roles with responsibilities, we propose to learn SVOs that are identifiable by agents' long-term behaviors by maximizing the conditional

[2] The reward-receiver agent cannot access these proportions, so it is impossible to reverse infer the reward giver's reward.

mutual information between the individual trajectory and the emergent SVO given the current observation and other agents' actions, which is similar as (Wang et al., 2020).

# 2 Preliminaries

Although studies on social dilemmas have contributed significantly to the research of cooperation emergence for decades (Axelrod and Hamilton, 1981; Peysakhovich and Lerer, 2018a; Anastassacos et al., 2020), they focus on matrix games and fixed binary policies. To be more realistic, as in real-world situations, the MARL community considers the intertemporal social dilemmas (ISDs, Leibo et al. (2017); Hughes et al. (2018b)). Before conducting numerical experiments, we first give the formal definition of ISD as follows. An ISD can be modeled as a partially observable general-sum Markov game (Hansen et al., 2004),

$$\mathcal{M} = \left\langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^{N}, \{\mathcal{O}_i\}_{i=1}^{N}, \mathcal{P}, \mathcal{E}, \{\mathcal{R}_i\}_{i=1}^{N} \right\rangle,$$

where $\mathcal{I}$ represents the $N$-agent space. $s \in \mathcal{S}$ represents the true state of the environment. We consider partially observable settings, where agent $i$ is only accessible to a local observation $o_i \in \mathcal{O}_i$ according to the emission function $\mathcal{E}(o_i \mid s)$. At each timestep, each agent $i$ selects an action according to a policy $a_i \in \pi_i\left(a \mid o_i\right)$, forming a joint action $\boldsymbol{a} = \langle a_1, \ldots, a_N \rangle \in \times \mathcal{A}_i$, results in the next state $s'$ according to the transition function $P\left(s' \mid s, \boldsymbol{a}\right)$ and a reward $r_i = \mathcal{R}_i(s, \boldsymbol{a})$. In ISDs, agents must learn cooperation or defection policies consisting of potentially long sequences of environmental actions instead of taking atomic cooperation or defection actions. In this paper, we focus on the episodic game with horizon $T$, and the goal of each agent is to maximize the *local* expected return, i.e.,

$$Q_i^{\pi}(s, \boldsymbol{a}) = \mathbb{E}_{s_{0:T}, \boldsymbol{a}_{0:T} \sim \pi, P}\left[\sum_{t=0}^{T} \gamma^t \mathcal{R}_i\left(s_t, \boldsymbol{a}_t\right) \mid s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}\right].$$

# 3 Methods

This section proposes a novel learning framework, RESVO, that transforms role-based learning into an SVO emergence problem to solve the ISD. Because consumers altruistically share a part of their profits with producers and making reward data unrestrictedly accessible is undesirable for several reasons, the proposed RESVO achieves SVO emergence by endowing agents with altruism to learn to share rewards with different weights to other agents. An SVO-based role embedding space is then constructed by introducing a novel low-rank constraint and conditioning individual policies on roles to ensure that the emergent SVO can effectively represent the different roles of agents and associate roles with responsibilities. Therefore, the following content in this section will be expanded from two aspects: SVO-based role emergence and role-based policy optimization.

## 3.1 SVO-based Role Emergence

As mentioned in Section 1, to consider the fact that consumers altruistically share a part of their profits with producers and avoid the realistic constraint (making reward data unrestrictedly accessible) imposed by directly learning the SVO of each agent according to the independence theory, RESVO enables agents to learn to dynamically share the reward with other agents, as shown in Figure 4. Specifically, the SVO-based role emergence mechanism learns an orientation function for each agent by explicitly accounting for its impact on recipients' behavior and, through them, the impact on its extrinsic objective. Each agent gives rewards using its orientation function and learns an SVO-conditioned policy with all received rewards. For clarity, we use index $i$ when referring to the reward-sharing part of an agent, and we use $j$ for the part that learns from the received reward, which is similar with Yang et al. (2020).

A reward-sharing agent $i$ learns a individual orientation function, $w_{\eta_i}^{i} : \mathcal{O}_i \times \mathcal{A}_{-i} \mapsto \mathbb{R}^N$, parameterized by $\eta_i$, that maps its own observation $o_i$ and all other agents' actions $a_{-i}$ to a vector of reward-sharing ratios for all $N$ agents. Unlike the existing methods based on SVO (Peysakhovich and Lerer, 2018b; Baker, 2020; Gemp et al., 2022; Yi et al., 2021) or saction (Koster et al., 2020; Lupu and Precup, 2020; Yang et al., 2020; Vinitsky et al., 2021; Dong et al., 2021) mechanism, the orientation function in RESVO **(1)** allows agents to reward itself [3], and **(2)** the sum of all sharing ratios does not need to be equal to 1. This is one of the reasons why reward sharing, a mechanism used by existing work, can encourage the division of labor and solve ISD. The intuition behind this lies in the particular properties of the public good dilemma. If there is a good division of labor among agents, the reward (punishment) of the agent does not come entirely from its behavior but partly from the producers (the consumers). Therefore, the agent that gets the reward needs to share a part with other agents and only gets a part of it (corresponding to the first point); Moreover, in a multi-agent scenario, a reward may come from the behavior of multiple producers, so it needs to share the same reward with multiple agents (corresponding to the second point).

Similar with Lupu and Precup (2020); Yang et al. (2020), $w_{\eta_i}$ is separate from the agent's conventional policy and is learned via direct gradient descent on the agent's extrinsic objective to reduce the learning difficulty. Specifically, at each timestep $t$, each recipient $j$ receives a total reward

$$r_j(\boldsymbol{\eta}, \boldsymbol{r}) := w^j_{\eta_j}[j] \cdot r_j + \sum_{i \neq j} w^i_{\eta_i}[j] \cdot r_i, \tag{1}$$

where $w^j_{\eta_j}[j]$ and $w^i_{\eta_i}[j]$ denotes the $j$-th elements of $w^j_{\eta_j}$ and $w^i_{\eta_i}$ respectively, $\boldsymbol{r} := [r_0, \cdots, r_N], \boldsymbol{\eta} = [\eta_1, \cdots, \eta_N]$. Although the sharers' rewards appear in Equation 1, the recipients can only see the sharers' discounted rewards when implemented. Each agent $j$ learns a SVO-based role conditioned policy $\pi_j(\cdot \mid o_j, e_j(\boldsymbol{\eta}))$ parameterized by $\theta_j$, where $e_j(\cdot)$ is the SVO-based role embedding of agent $j$. After each agent has updated its policy to $\hat{\pi}_j$, parameterized by new $\hat{\theta}_j$, with role-based policy optimization (Section 3.2) via trajectories $\tau_i$ sampled by joint policies $\{\pi_j\}$, we sample a set of new trajectories with new joint policy $\{\hat{\pi}_j\}$. Using these trajectories, each agent $i$ updates the individual orientation parameters $\eta_i$ to maximize the following objective

$$\max_{\eta_i} J^{\text{svo}}(\hat{\tau}_i, \tau_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}) := \mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\sum_{t=0}^{T} \gamma^t \hat{r}^t_i\right], \quad \text{s.t. } \operatorname{rank}(W^t_{\boldsymbol{\eta}}) = k, \forall t \in [0, T), \tag{2}$$

where $\hat{r}^t_i$ is the newly sampled extrinsic reward in $\hat{\theta}_i$, $W^t_{\boldsymbol{\eta}} = \{w^{i,t}_{\eta_i}\}^N_{i=1}$ is the matrix composed of the reward sharing ratios of all agents at timestep $t$, and $k \leq N$ is a hyperparameter. To ensure that the emergent SVO can effectively represent the different roles of agents, RESVO introduces a novel rank constraint on the SVO matrix $W^t_{\boldsymbol{\eta}}$ of all agents, and $k$ can be regarded as the theoretical optimal number of roles. To be able to optimize (2) with an automatic differentiation toolkit in an end-to-end manner, we transform (2) into the following unconstrained optimization problem based on projected gradient descent by introducing an intrinsic reward

$$\max_{\eta_i} J^{\text{svo}} := \mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\sum_{t=0}^{T} \gamma^t \left(\hat{r}^t_i - \alpha \| W^{i,t}_{\eta_i} - W^{i,t}_k \|^2_2\right)\right], \tag{3}$$

where $W^t_k$ is the $k$-rank approximation of $W^t_\eta$ obtained with SVD algorithm and $\alpha^t$ is another hyperparameter, and superscription $i$ denotes the $i$-th column of the matrix. In practice, following the derivation process of (Yang et al., 2020), one can define the loss as

$$\begin{aligned} -\sum_{t=0}^{T} \sum_{j=1}^{N} \log \pi^j_{\hat{\theta}^j} \left(\hat{a}^t_j \mid \hat{o}^t_j, \hat{e}^t_j(\boldsymbol{\eta})\right) \cdot \\ \sum_{\ell=t}^{T} \gamma^{\ell-t} \left(\hat{r}^\ell_i - \alpha \| \Delta^{i,\ell}(W, k) \|^2_2\right) - 2\alpha \nabla_{\eta_i} W^{i,t}_{\eta_i} \Delta^{i,t}(W, k), \end{aligned} \tag{4}$$

[3] But, this does not mean that the orientation function will converge to some trivial function, such as the agent giving itself an infinite reward. Because the final reward received by the agent is its reward multiplied by the sharing ratio, and the sharing ratio is in the closed range of 0 to 1.

and directly minimize it via automatic differentiation, where $\Delta^{i,\ell}(W,k) := W^{i,\ell}_{\eta_i} - W^{i,\ell}_{k}$ and $\Delta^{i,t}(W,k) := W^{i,t}_{\eta_i} - W^{i,t}_{k}$. Crucially, $\hat{\theta}^j$ must preserve the functional dependence of the policy update step (7) on $\eta_i$ within the same computation graph.

It is worth noting that the agent's role representation $e_i(\boldsymbol{\eta})$ is **NOT** composed of its reward *sharing* ratios $w^i_{\eta_i}$. However, its reward *recipient* ratios, i.e., we take the $i$-th row in the matrix $W^t_{\boldsymbol{\eta}}$ as the role representation of agent $i$ at timestep $t$, as shown by interdependence theory in Figure 1. Intuitively, agents with similar role representations have similar divisions of labor and thus receive similar rewards. In other words, RESVO decomposes the extrinsic reward received by all agents into several parts according to the functional composition of the agent during the SVO emergence learning stage. The difference in rewards will directly lead to differences in the agents' policies, thereby encouraging the formation of the division of labor among the agents.

## 3.2 Role-based Policy Optimization

Introducing SVO-based role embedding and conditioning individual policies on this embedding explicitly establishes the connection between the role and the individual policies to encourage the division of labor through the diversity of roles. However, this does not enable the role to constrain the agent's long-term behavior. Intuitively, conditioning roles on local observations and actions[4] enables roles to be responsive to the changes in the environment but may cause roles to change quickly. Thus roles and responsibilities cannot be effectively associated. To address this problem, we expect SVO-based roles to be temporally stable.

Drawing inspiration from Eysenbach et al. (2019); Wang et al. (2020), we propose to learn SVO-based roles that are identifiable by agents' long-term behaviors, which can be achieved by maximizing $I(\tau_i; e_i \mid \boldsymbol{o}, \boldsymbol{a})$, the conditional mutual information between the individual trajectory and the role given the current *joint* observation and *joint* action. The conditional mutual information calculated here is based on the joint actions and observations of all agents because $e_i$ cannot be generated by local observations $o_i$. The role representation of agent $i$, $e_i$, is the $i$-th **row** in the matrix $W_{\boldsymbol{\eta}}$. Based on variational inference, a variational posterior estimator can be proposed to derive a tractable lower bound for the mutual information objective

$$\max_{\boldsymbol{\eta},\phi} J_i^{\mathrm{mi}} := I(e_i^t; \tau_i^{t-1} \mid \boldsymbol{o}^t, \boldsymbol{a}^t) \geq \mathbb{E}_{e_i^t, \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log \frac{q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)}{W_{\boldsymbol{\eta}}\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)}\right], \tag{5}$$

where "mi" stands for "mutual information", $\tau_i^{t-1} = \left(o_i^0, a_i^0, \cdots, o_i^{t-1}, a_i^{t-1}\right)$[5] and $q_\phi$ is the variational estimator parameterized by $\phi$. Here we use $W_{\boldsymbol{\eta}}$ to represent orientation function $w^i_{\eta_i}$ of all agent $i$. For $q_\phi$, we use a causal transformer (Chen et al., 2021) (shared by all agents) to encode an agent's history of observations and actions. The lower bound in (5) can be further rewritten as a loss function to be minimized via automatic differentiation

$$\begin{aligned} \mathrm{L}^{\mathrm{mi}}\left(\boldsymbol{\tau}; \boldsymbol{\eta}, \phi\right) &= \frac{1}{n} \sum_{i=1}^{N} \mathrm{L}_i^{\mathrm{mi}}\left(\tau_i; \boldsymbol{\eta}, \phi\right) \\ &= \mathbb{E}\left[D_{\mathrm{KL}}\left[W_{\boldsymbol{\eta}}\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right) \| q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right]\right], \end{aligned} \tag{6}$$

where $D_{\mathrm{KL}}[\cdot\|\cdot]$ is the KL-divergence operator, and the conditioned terms in $p, q_\phi$ are omitted for convenience. See the Appendix for the detailed derivation.

In addition, to further improve learning stability, the individual policies are no longer based on the current role embedding but on the latest $m$ role embeddings to make decisions. That is, each agent $j$ learns a SVO-based role conditioned policy $\pi_j(a_j^t \mid o_j^t, \{e_j^l(\boldsymbol{\eta})\}_{l=t-m+1}^t)$ parameterized by $\theta_j$. Our experiments found that for complex ISDs with long episodes, mutual information constraints and decision-making based on historical role embeddings can be crucial in improving performance. Thus, the objective of role-based policy

[4] The orientation function $w^i_{\eta_i}$ maps agent $i$'s observation $o_i$ and all other agents' actions $a_{-i}$ to a vector of reward-sharing ratios for all $N$ agents.

[5] There's a bit of misuse of notation here, and compared to Section 3.1, no rewards are included in the trajectory here.

optimization for each agent $j$ is shown as follows based on the multi-agent policy gradient theorem (Lowe et al., 2017)

$$\max_{\theta_j} J^{\text{policy}}(\theta_j, \boldsymbol{\eta}) := \mathbb{E}_{\boldsymbol{\pi}(\cdot|\cdot, e_j(\boldsymbol{\eta}))}\left[\sum_{t=0}^{T} \gamma^t r_j^t(\boldsymbol{\eta}, \boldsymbol{r}^t)\right]. \tag{7}$$

## 3.3 Algorithm Summary

The learning process of RESVO consists of two main steps, namely, SVO-based role emergence (Eq. 2) and role-based policy optimization (Eq. 7). To improve the expressiveness and responsibility of role representation, we additionally impose rank constraints (Eq. 2) and mutual information regularization (Eq. 5) to the objectives of role emergence and policy optimization. Therefore, the final learning objective of RESVO for each agent $i$ is

$$\max_{\theta_i, \eta_i, \phi} J := \overbrace{\lambda_{\text{svo}} J^{\text{svo}} + \lambda_{\text{mi}} J^{\text{mi}}}^{\text{role emergence}} + \underbrace{\lambda_{\text{p}} J^{\text{policy}}}_{\text{policy optimization}}, \tag{8}$$

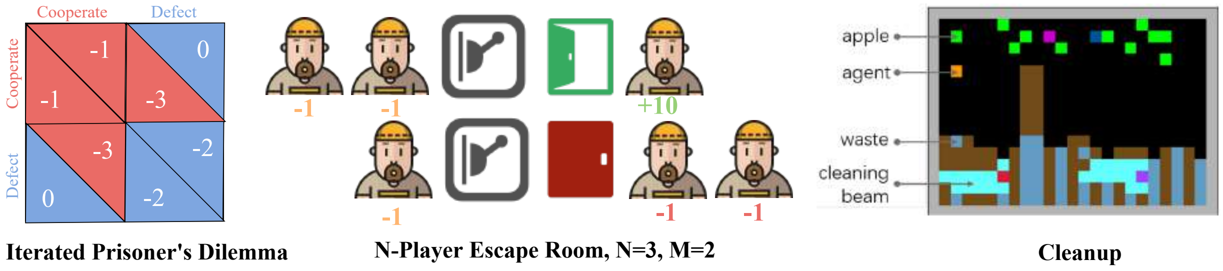
where $\lambda_{\text{svo}}, \lambda_{\text{mi}}, \lambda_{\text{p}}$ are scaling factors. Furthermore, the pseudo-code of the proposed RESVO is shown in Algorithm 1.

**Algorithm 1** Leanring Role with Emergent SVO (RESVO)

```
Require: Initialize parameters of policies $\theta_i$, orientation function $\eta_i$, variational estimator $\phi$;
for each iteration do
    Collect interaction $\tau_i$ with $\pi_{\psi_i}$ and update replay buffer $\mathcal{D} \leftarrow \{\tau_i\}$;
    Generate trajectories $\{\tau_i\}$ using $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$, and for all reward receivers $j$, update $\hat{\theta}_j$ via (7);
    Generate new trajectories $\{\hat{\tau}_i\}$ using new $\hat{\boldsymbol{\theta}}$ and for reward sharers $i$, compute $\hat{\eta}_i$ via (2),(5);
    For variational estimator, compute $\hat{\phi}$ via (5), $\theta_i \leftarrow \hat{\theta}_i$, $\eta_i \leftarrow \hat{\eta}_i$ for all $i$, $\phi \leftarrow \hat{\phi}$.
end for
```

# 4 Experiments

Although studies on social dilemmas have contributed significantly to the research of cooperation emergence for decades (Axelrod and Hamilton, 1981; Peysakhovich and Lerer, 2018a; Anastassacos et al., 2020), they focus on matrix games and fixed binary policies. To be more realistic as in real-world situations, the MARL community considers the intertemporal social dilemmas (ISDs, Leibo et al. (2017); Hughes et al. (2018b), see Appendix 2 for the formal definition) which can be modeled as a partially observable general-sum Markov game (Hansen et al., 2004).



Iterated Prisoner's Dilemma

N-Player Escape Room, N=3, M=2

Cleanup

Figure 5: Three different environments with increasing complexity.

Our experiments demonstrate that the division of labor can effectively solve the ISDs of different difficulties. Moreover, compared with other mechanisms, the division of labor can converge to mutual cooperation faster and more stably. Our method is tested in the following three typical public good dilemmas with increasing complexity (see Figure 5) against several baselines.

- The first one, Iterated Prisoner's Dilemma (IPD) (Foerster et al., 2018), is often regarded as the canonical and most difficult of 2-by-2 game-theoretic cooperation problems. The prisoner setting may seem contrived, but there are, in fact, many examples of human interaction as well as interactions in nature that have the same payoff matrix. Many natural processes have been abstracted into models in which living beings are engaged in endless games of prisoner's dilemma, e.g., the climate-change politics in environmental studies (Rehmeyer, 2012), the reciprocal food exchange of vampire bats (Davis, 2017), the doping in sport (Schneier, 2012) and the coherence of strategic realism in the international political (Majeski, 1984). This broad applicability of the PD gives the game substantial importance. In an IPD, agents observe the joint action taken in the previous round and receive rewards in Figure 5. Although, by definition, a public good dilemma needs to contain more than 2 agents (Kollock, 1998), the prisoner's dilemma can be viewed as a simplified version. The agent that selects the "cooperate" action corresponds to the "producer", and the agent that selects the "defect" corresponds to the "consumer". In this simplified version of the public good dilemma, defect or free-riding is the dominant strategy.

- The second one, $N$-Player Escape Room (ER) (Yang et al. (2020, Figure 1)), is a discrete $N$-player Markov game with parameter $M < N$. ER is a more complex public good dilemma. Since $M = 2$, $N = 3$, the optimal joint policy of the game has an asymmetric division of labor, which is also widespread in the economic activities of human society. Specifically, there are 3 discrete states in this game, i.e., "start", "door" and "lever". An agent gets $+10$ extrinsic reward for exiting a "door" and ending the game, but the "door" can only be opened when $M$ other agents cooperate to pull the "lever". However, an extrinsic penalty of $-1$ for any movement discourages all agents from taking this cooperative action.

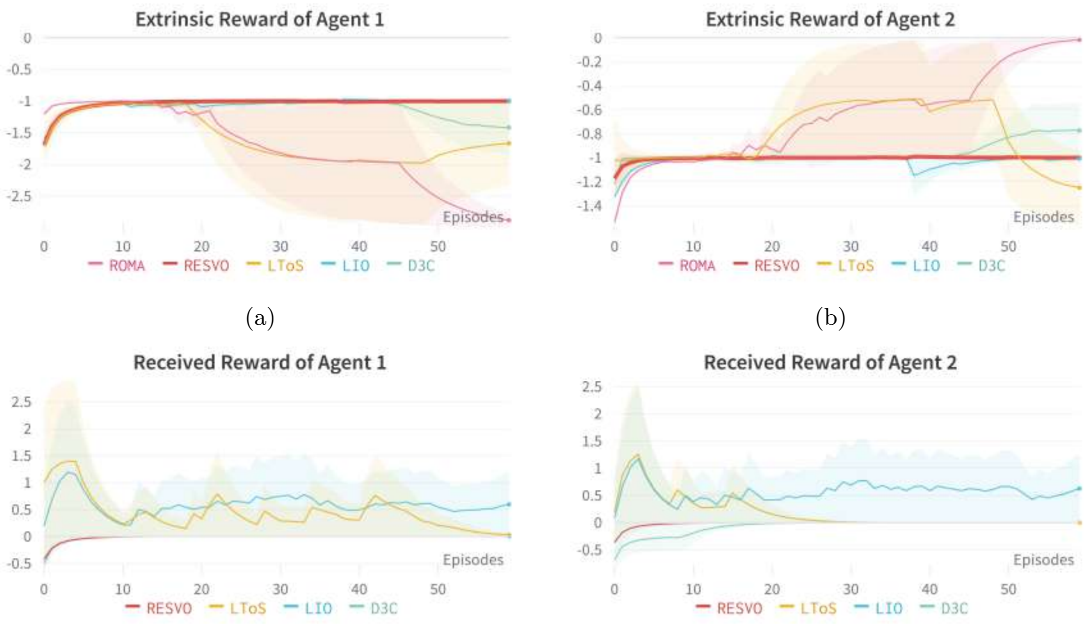
- In the third one, Cleanup (Leibo et al., 2017; Hughes et al., 2018b) is a high-dimensional grid-world intertemporal social dilemma that serves as a problematic benchmark for independent learning agents. Moreover, the Cleanup can also be considered as a simplified version of the tax simulator, *Gather-Trade-Build (GTB)* (Zheng et al., 2022), but since there is no significant division of labor in the tasks involved in the latter, it is not considered in this paper. In the Cleanup, agents get $+1$ (for the small 10 by 10 map) or $+0.25$ (for the big 48 by 18 map) reward by collecting apples, which spawn on the map at a linear decay rate as the amount of waste approaches a depletion threshold. Each episode starts with a waste level above the threshold and no apple present. An agent can contribute to the public good by firing a cleaning beam to clear waste (no reward). This would enable other agents to be free riders, resulting in a problematic public good dilemma.

We selected the aforementioned role learning algorithm, **ROMA** (Wang et al., 2020), and sanction-based algorithm, **LIO** (Yang et al., 2020), as the baselines because our RESVO draws on the core ideas of them in the SVO-based role emergence and the role-based policy optimization[6] respectively. In addition, we selected several aforementioned SVO-based algorithms, including **LToS** (Yi et al., 2021) and **D3C** (Gemp et al., 2022) as baselines. Below we will briefly introduce the core idea of each algorithm again. ROMA designs an end-to-end specialization learning objective to encourage the emergence of roles in general MARL tasks for better cooperation and generalization and avoid the requirement of prior or expert knowledge. LIO enables each agent to punish, thereby implementing the sanction mechanism in a decentralized manner. LToS uses a bi-level optimization scheme similar to LIO but uses meta-gradients instead of data sampled by other agents' updated policies for the learning of SVOs; D3C uses the price of anarchy instead of the joint expected cumulative reward in LToS as the optimization objective of SVOs. Meanwhile, the SVO-based role emergence mechanism in RESVO can be introduced into other SVO-based methods as a plug-and-play module. In order to verify the general promotion effect of the division of labor on SVO-based methods, we

[6] See Method (Section 3) for more details.

added the role learning mechanism in RESVO to two SVO-based works, denoted as **LToS+r** and **D3C+r** respectively.

## 4.1 Role Emergence in the Classic Tasks



(a)

(b)

(c)

(d)

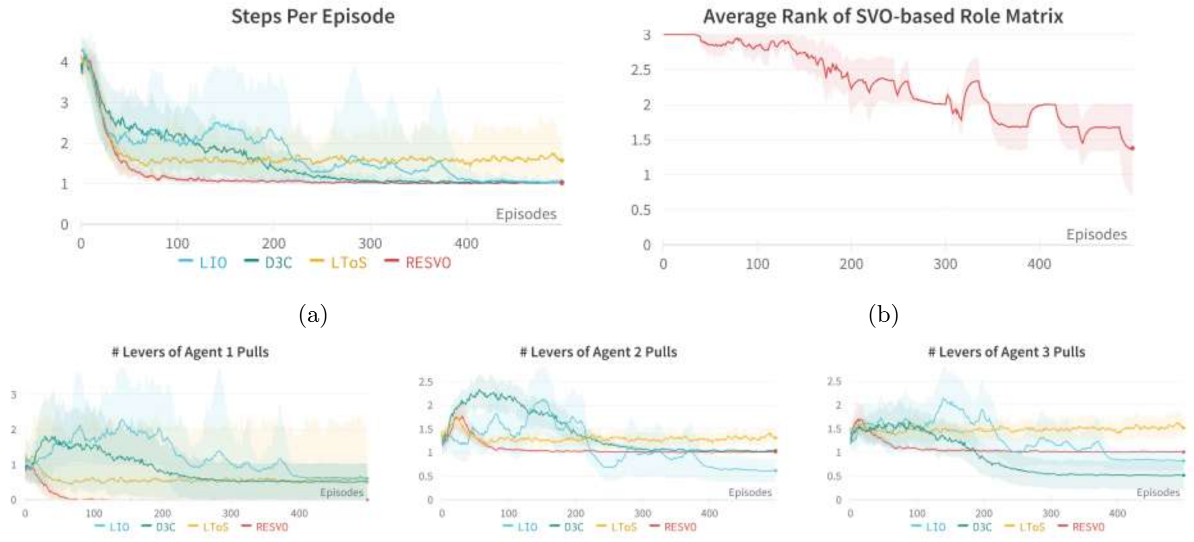
Figure 6: Extrinsic (a-b) and received rewards (c-d) of different algorithms in Iterated Prisoner's Dilemma. Since ROMA is not designed based on SVO, rewards will not be transmitted between agents.

We first analyze the performance of each algorithm on the 2-players task, IPD. Figure 6 shows the performance and emerged SVOs of RESVO and the performance of four baselines in the IPD environment. In the IPD, a simplified version of the public good dilemma, we want both agents to be producers (that is, to cooperate with each other) to achieve the highest social welfare. Therefore, we set the rank constraint in RESVO to 1. This shows that the optimal joint policy of IPD requires only one role: the producer or the cooperator. It can be seen from the figure that the ROMA algorithm based only on the division of labor cannot stably converge to mutual cooperation. Both players choose the cooperate action and receive a $-1$ environmental reward. In the early stages of training, ROMA can learn to cooperate. Nevertheless, once a player chooses the "defect" action, ROMA's role learning will fix the player's character, so she will always choose the "defect" action and make mutual cooperation unsustainable.

The LToS and D3C algorithms based only on the SVO mechanism can converge to mutual cooperation quickly, but this equilibrium also cannot be maintained for a long time. However, the reasons for the inability of these two baselines to maintain mutual cooperation are not the same. The LToS algorithm is that because the agent does not form a stable SVO, agent 1 no longer shares rewards with agent 2 from the middle of training (Figure 6d), so that agent 2 begins to explore other strategies. Although the D3C algorithm forms a stable SVO, the agent's policy and SVO have no conditional dependence, causing the policy to diverge at the end of the training. The LIO algorithm and the RESVO algorithm show the best results. However, LIO still has performance fluctuations after a training period and cannot maintain stable mutual cooperation. This suggests that the sanction-based approach does not sustain stable cooperation in IPD.

Moreover, an interesting phenomenon can be found in Figure 6(c-d), where RESVO and other SVO-based baselines (i.e., LToS and D3C) learn two completely different SVOs to try to maintain cooperation.

As seen from the figure, after RESVO converges to mutual cooperation, the agent no longer needs to receive rewards, nor does it share rewards. However, agents trained by other baselines have always received and shared rewards during the training process. This illustrates that while RESVO maintains cooperation through individualism orientation, other baselines achieve the same result through martyrdom orientation, where the coefficients in the transformation matrices are all negative[7]. Our analysis suggests that another reason for the inability of LToS and D3C to maintain stable cooperation may be due to this martyrdom orientation, an SVO that causes rewards to be passed between agents all the time. After policies converge to the equilibrium of mutual cooperation, these rewards will become noise and cause instability in the training process. In contrast, RESVO shifts from a cooperative orientation to an individualism orientation after the policies reach equilibrium, thus making the equilibrium remain stable. Although LIO is not an SVO-based method, the sanction mechanism also allows reward transmission between agents all the time, thus leading to the same instability of the training process.



(a)

(b)

(c)

(d)

(e)

Figure 7: (a) Steps per episode. (b) The average rank of the SVO-based role matrix. (c-e) The number of levers each agent pulls of different algorithms in 3-Player Escape Room.

Next, we expand our study scenario into a more complex 3-player Escape Room. Figures 7 shows the performance of RESVO and baselines in the 3-Player Escape Room with $M = 2$. This means that two agents must pull the lever for the agent to open the door. Figures 7(b) shows the satisfaction of the rank constraints of the RESVO algorithm during training. In the 3-player Escape Room, we set the rank constraint of RESVO to 2. Specifically, we hope 2 roles emerge from the three agents through SVO, the lever-puller, and the door-opener. As can be seen from the figure, as the training progresses, the rank constraint of the RESVO algorithm can be well satisfied.

A simple analysis of this task shows that the optimal policy only requires each agent to perform one-step action. This is because it only takes one timestep for any agent to move from the “initial” to the “lever” or “gate”. At the same time, each move will bring a reward of $-1$, and a more than 1 move will be a social welfare decline. The Figure 7(a) records the steps required by different algorithms to complete the task. Since ROMA performs poorly on the most straightforward IPD task, we no longer show the performance of ROMA on the more complex Escape Room and Cleanup tasks. It can be seen that RESVO can converge to the optimal policy fastest. D3C and LIO take about 4 times as many samples to converge to equilibrium compared to RESVO. LToS falls into a locally optimal solution early and cannot escape from it.

[7] In IPD, the external rewards of the agents are all negative, so only negative coefficients can deliver positive rewards.

Figure 7(c-e) shows the division of labor of different agents in the 3-Player Escape Room with different algorithms. It can be seen from these 3 figures that whether the division of labor is formed and whether the division of labor is stable dramatically affects the performance and convergence speed of the algorithm. RESVO has converged to a stable division of labor: agent 1 opens the door, and agents 2 and 3 pull the lever. Therefore, RESVO can converge to equilibrium the fastest and maintain it compared with baselines.

For the D3C algorithm, agent 2 is stably assigned the role of a lever-puller, but there is no stable division of labor between agent 1 and agent 3. The two oscillate between the roles of lever-puller and door-opener. None of the three agents have their own fixed roles for the LIO algorithm. Moreover, comparing Figure 7(a) and Figure 7(c-e), it can be seen that even after the LIO algorithm converges to equilibrium, the three agents are still dynamically allocated between the two roles of lever-puller and door-opener. The instability of the division of labor between D3C and LIO also affects their convergence speed, and LIO converges more slowly than D3C. For the LToS, the three agents have been unable to form a correct division of labor, and the average number of lever pulls greater than twice, indicating that the average number of the lever-puller is less than 2. This will make an agent need to change from the “door-opener” to the “lever-puller” to complete the task successfully. The role change process mentioned above makes the timestep of the LToS algorithm to complete the task more significant than 1 on average.

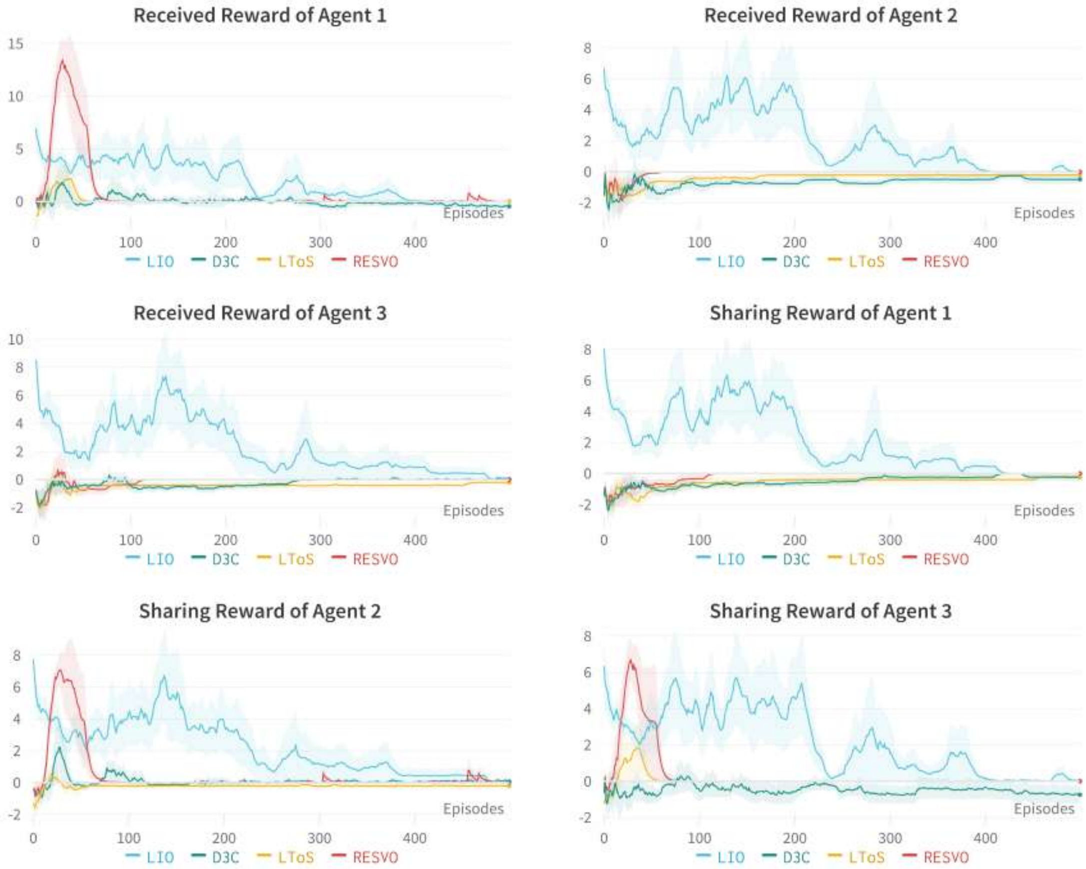


Figure 8: Shared and received rewards of different algorithms 3-Player Escape Room.

In order to explore the reason why RESVO can form a stable division of labor, we show in Figure 8 how different agents share and receive rewards in different algorithms. That is, the emerged SVOs of three SVO-based methods (including RESVO) and LIO. It is worth noting that in more complex Escape Room and Cleanup environments, in order to make SVO have powerful representation capabilities, RESVO uses the transition matrix coefficients of multiple consecutive time steps as SVO. This differs from the setting in our IPD experiments, making it impossible to classify the agent as a certain kind of SVO. Therefore, we indirectly analyze the agent's SVO from the pattern of sharing rewards in the following.

A counter-intuitive phenomenon can be seen in the figure. Those who pull the lever have no profit, and those who open the door can obtain more significant benefits. Therefore, intuitively, to maintain a stable division of labor, the door opener should share her reward with the lever-puller so both parties can get rewards. At the same time, since each agent is self-motivated, this can form a stable role division. However, the three SVO-based algorithms, RESVO, LToS, and D3C, share the rewards in turn. Those who pull the levers give rewards to those who open the door. The difference between these three algorithms is the size of the reward shared. A plausible explanation is that the SVO-based algorithm learned a more "aggressive" approach. The agent that discovers the action of pulling the lever gives the agent that discovers the action of opening the door a positive reward so that the role of the door opener is fixed. Then determine the role of the lever-puller. This method can instead promote the rapid formation of the division of labor and regional stability for RESVO. The rank constraints in RESVO and the policy conditioned on SVO enable another role can also be fixed once a role is fixed. Nevertheless, other SVO-based methods do not have this advantage, so the algorithm converges slowly during the training process, and the division of labor cannot be maintained stably. It can be seen from the figure that at the beginning of training, RESVO agents share a large-scale reward value, thus promoting the algorithm to converge quickly. After the algorithm converges, similar to the IPD task, the agent almost no longer shares and receives rewards, thus maintaining the stability of the policies and division of labor.

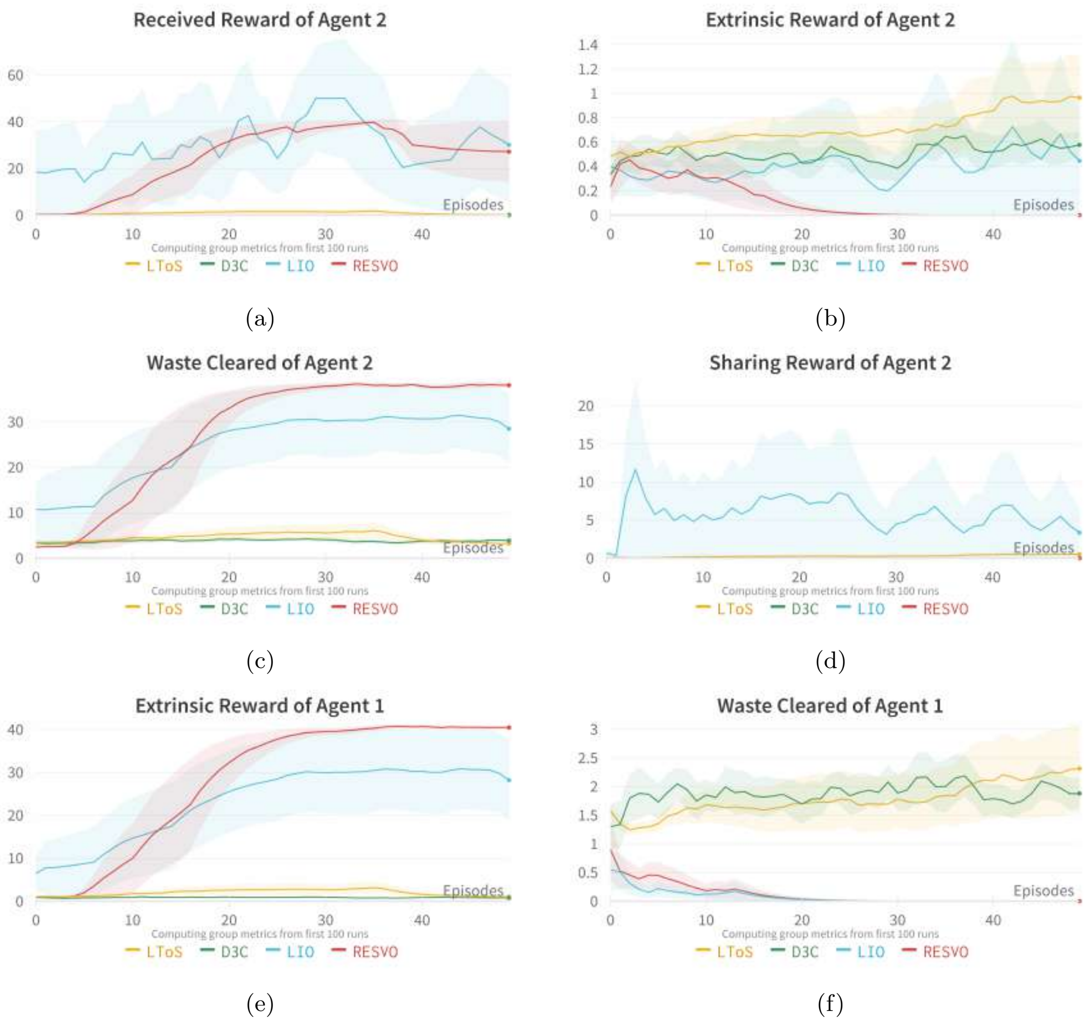
The sanction-based LIO method exhibits a similar pattern to that in the IPD environment. During the entire training process, even after the policies converges to equilibrium, the agents continue to transmit large rewards, maintaining the stability of the division of labor through a high “cost” way. One defect of this method is that the module that generates the sanction will always receive a large gradient during the training process, which makes the optimization process unstable. The algorithm converges slowly and cannot maintain a stable division of labor for a long time.

## 4.2 Role Emergence in the Cleanup

We finally test our method in Cleanup and start from a map size of 10 by 10 and the number of agents 2. Although there are only 2 agents, it is still much more complicated than $N$-Players Escape Room because Cleanup has far more timesteps per episode than the former. In this task, similar to the 3-Player Escape Room, there is also an apparent division of labor between the two agents under the optimal cooperative policy: one agent needs to clean up wastes (producer), and the other agent needs to collect apples (consumer). We can judge the division of labor or roles of the two agents from the amount of waste they clean up.

As seen from Figure 9, different algorithms also show remarkable differences in the 2-agents Cleanup task. On the one hand, LToS and D3C have not learned a good division of labor, and both agents hardly clean up waste (Figure 9(c) and (f)), so no apples grow. This makes the extrinsic reward for both agents small (Figure 9(b) and (e)), and the joint policy suffers from a public good dilemma. As can be seen from Figure 9(a) and (d), the two agents hardly share rewards, and their SVOs are almost the same, which is why LToS and D3C cannot learn a good division of labor or roles. On the other hand, RESVO and LIO have successfully formed a reasonable division of labor. Agent 1 is responsible for collecting apples (consumer), and agent 2 is responsible for cleaning up wastes (producer, see Figure 9(c) and (f)). At the same time, to maintain a stable division of labor or roles, agent 1 as a consumer will continue to share rewards with agent 2 as a producer (Figure 9(a) and (d)), to achieve a larger average extrinsic reward, or the social welfare. That is, agent 2 maintains a stable division of labor by showing a cooperative orientation to agent 1.

However, similar to the previous tasks, the sanction-based LIO method is very different from RESVO in the way that roles emerge. The two algorithms exhibit different robustness in maintaining the division of labor. As can be seen from Figure 9(d), on the one hand, for LIO, agent 2, which is the producer, also needs to share the reward with agent 1. RESVO, on the other hand, uses a more "energy-efficient" approach to promoting role emergence: Since producers receive no rewards, there is no need to share rewards with consumers who can receive large extrinsic rewards. This sparsity of reward sharing between agents also



(a)

(b)

(c)

(d)

(e)

(f)

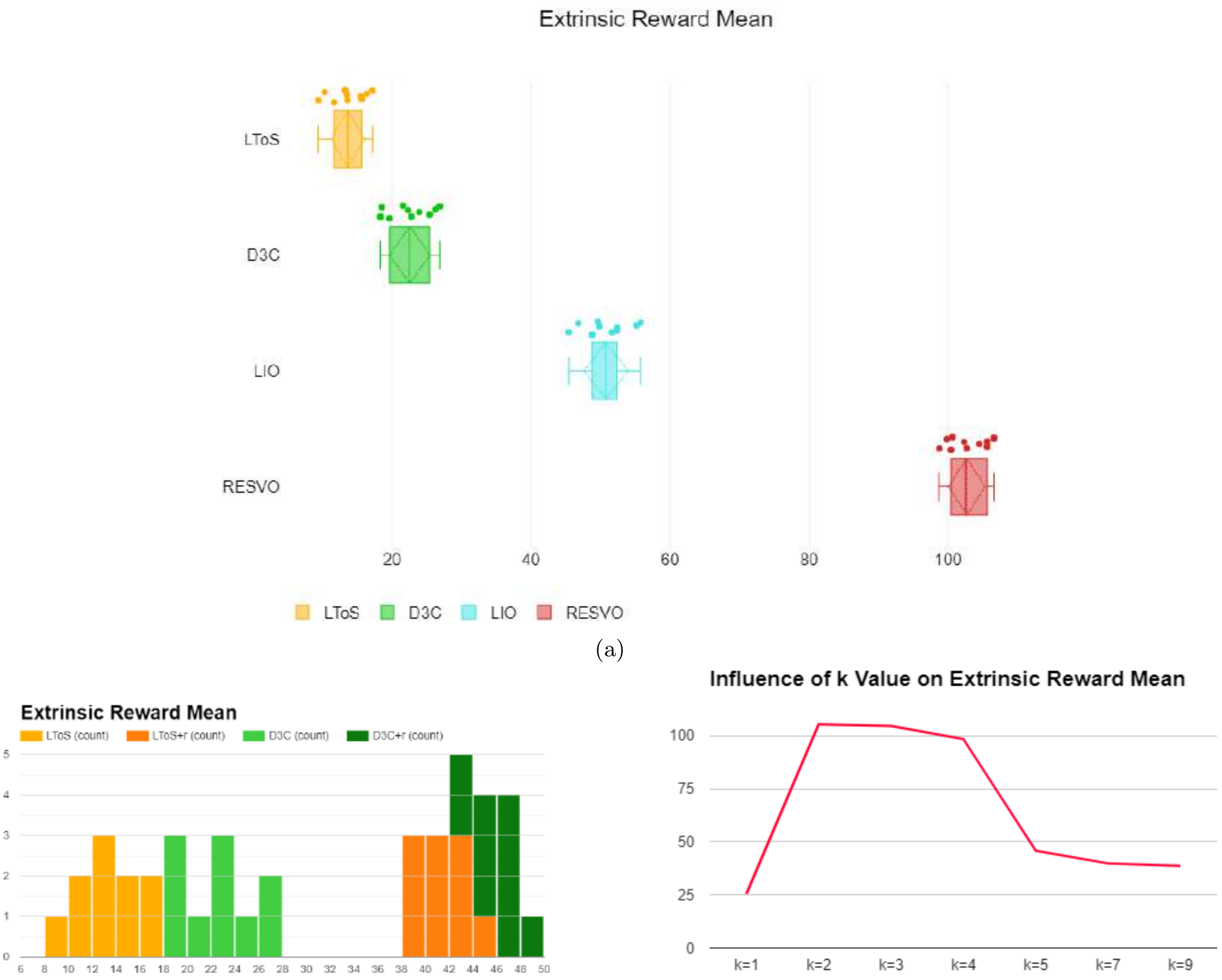
Figure 9: Shared and received rewards, extrinsic rewards, and waste cleared of different algorithms in $10 \times 10$ map size, 2-Player Cleanup.

enables RESVO to maintain a more stable division of labor while receiving greater social welfare. It can be seen from Figure 9(c) that the agent 2 trained by the LIO algorithm does not always clean up the waste, and its behavior shows a significant variance. This also makes the external reward of agent 1 unable to maintain a high level all the time and also has a significant variance (Figure 9(e)).

## 4.3 Static versus Dynamic Division of Labor

In the three tasks in the previous two sections, we find that the sanction-based LIO can also effectively learn a reasonable division of labor or roles. However, due to the difference between the sanction mechanism and SVO, LIO needs to transmit rewards densely between agents, which makes it impossible to maintain a stable division of labor. In other words, RESVO realizes a *static* division of labor through the emergence of SVO, but LIO realizes a *dynamic* division of labor. In the static division of labor, the role of each agent is fixed while completing the task; on the contrary, the agent's role will change in the dynamic division of labor.

In addition, from the experiments in the previous two sections, we find preliminary evidence that static division of labor can lead to better social welfare. Nevertheless, the above tasks only contain $2-3$ agents,



(a)

(b)

(c)

Figure 10: The (a) boxplot, (b) histogram of performance, and the (c) influence of $k$ value for extrinsic rewards under 10 runs of different algorithms in $48 \times 18$ map size, 10-Player Cleanup.

and the impact of dynamic division of labor cannot be fully reflected. For example, for the IPD or Cleanup task that only consists of 2 agents, the dynamic division of labor is the role reversal of the two agents. To this end, we compare the performance of the algorithms in a larger Cleanup environment with a map size of 48 by 18 and many agents of 10. There are only two roles in the Cleanup: the waste cleaner and the apple picker. When the number of agents exceeds 2, multiple agents will have the same role. Cooperation among roles and agents with the same role is required to get rid of the public good dilemma and achieve greater social welfare. At this time, for the cooperation of the same role, the static role or division of labor has more advantages. Because dynamic roles involve role changes, that is, the composition of group members with the same role changes. This will make the cooperation of the same role unstable, affecting the algorithm's performance. The larger the number of agents, the more severe the problem of unstable cooperation will become, which will also cause more significant performance degradation. To test the above hypothesis, we conducted multiple random experiments in 10-player Cleanup to ensure the reliability of the results. As seen from Figure 10, RESVO shows a clear performance advantage (about two times) compared to LIO in the more complex Cleanup environment.

Figure 10 verifies the impact of the dynamic division of labor in the task completion process of different algorithms on performance. We also find that the dynamics of the division of labor are not only reflected in the completion of one task but also in the completion of different tasks by recording the dynamics of labor division of different algorithms under different random seeds. Specifically, for each random experiment, we count the number of waste cleaned by each agent, similar to Figure 9(c) and (f). If the agent is cleaning a low amount of waste (closer to 0), the counter for the agent's role as a cleaner is incremented by one. Otherwise, the picker counter is incremented by one. Figure 11 shows the average dynamic where 10 agents are assigned the role of cleaning wastes by different algorithms under 10 random seeds. As seen from the figure, in different random experiments, the SVO-based methods, LToS, D3C, and RESVO, all show better static division of labor than the sanction-based LIO. However, LToS and D3C converge to a suboptimal static division of labor. Most of the agents are assigned the role of collecting apples. Both LIO and RESVO converge to a better division of labor. More agents choose to clean up waste, but the former maintains its dynamics. Combined with the results of Figure 10, it can be seen that in tasks involving more complex optimal division of labor patterns, the static division of labor learned by RESVO can be more efficient than the dynamic division of labor in LIO.

To more intuitively demonstrate the policies learned by different algorithms in the 10-player Cleanup, we selected keyframes from the rendered results, as shown in Figure 12 and 13. The arrow in the figure indicates the reward sharing. The policies shown in the figure match the performance results and related analysis presented earlier.

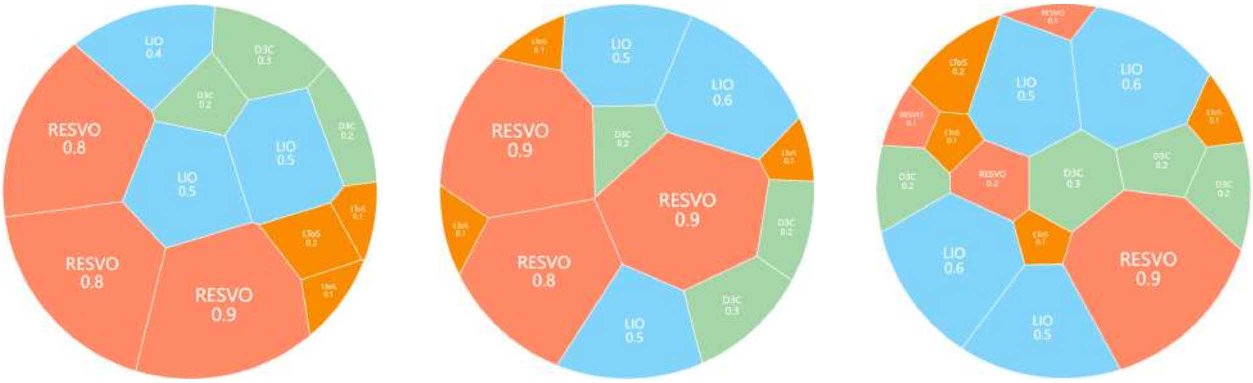


Figure 11: The role composition of different algorithms in 48 × 18 map size, 10-Player Cleanup. The numbers in the color blocks represent the number of times in 10 experiments assigned to cleaning up wastes. The size of the color block is proportional to the number.

## 4.4 SVO Emergence as a Plug-and-Play Module

As introduced in Introduction and Method, RESVO is based on the independence theory, which transforms the learning of social value orientation, i.e., the learning of transformation matrix, into a symmetric learning to share problem. In this way, the role representation of the agent is transformed from the transformation matrix to the sharing coefficient matrix. To constrain the number of emergent roles, RESVO introduces a novel rank constraint. At the same time, to make the agent's behavior subject to role constraints, RESVO introduces a conditional policy based on role representation and a mutual information constraint to accelerate the learning of this augmented policy. The above core idea of RESVO shows that the rank constraint and the conditional policy to satisfy the mutual information constraint are the two main modules of RESVO. Both modules are common to other SVO-based methods. Therefore, we can naturally add them to the SVO-based baselines, namely LToS and D3C, i.e., The SVO-based role emergence mechanism in RESVO can be introduced into other SVO-based methods as a plug-and-play module.

The numerical experimental results of LToS and D3C with the addition of the rank constraint and the

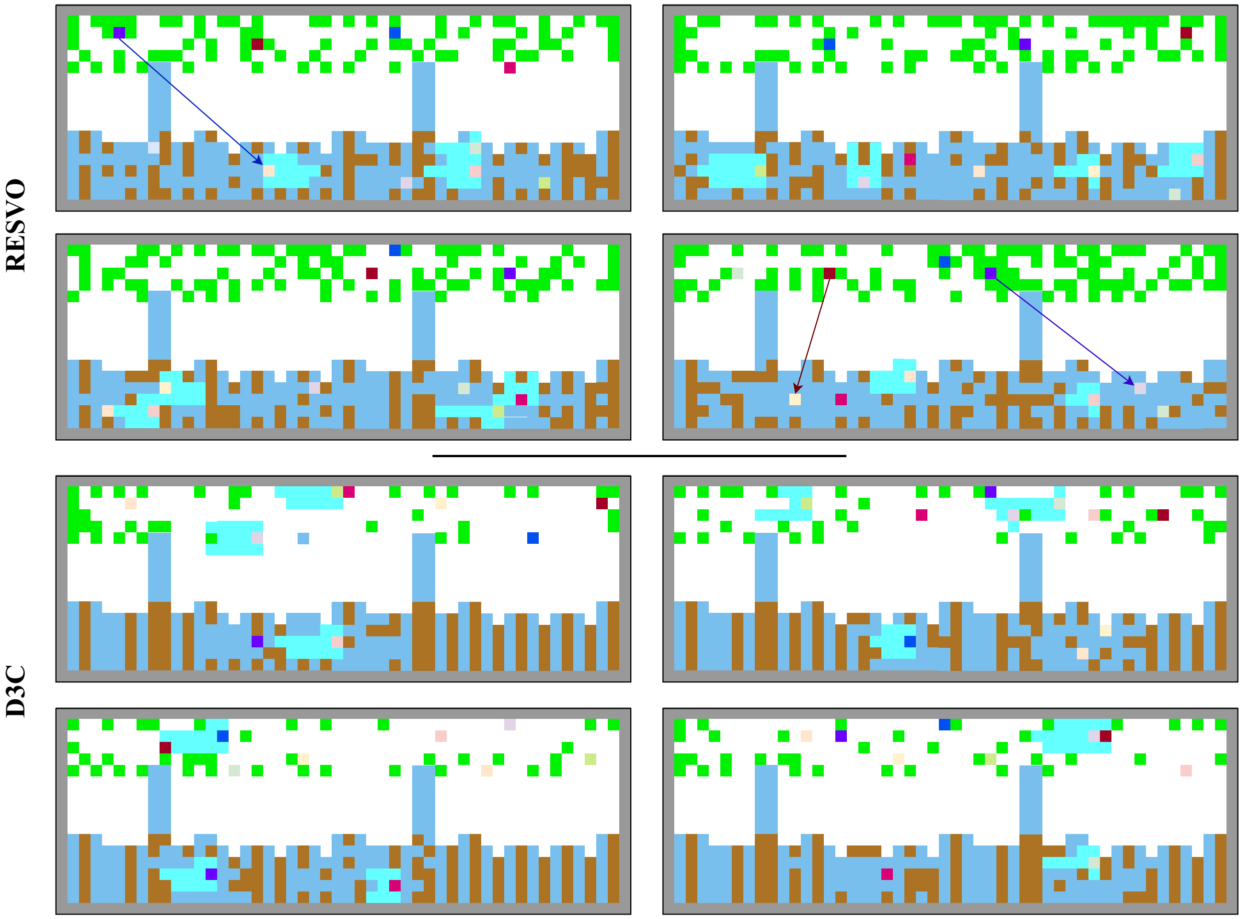


Figure 12: Selected keyframes from the rendered results of RESVO and D3C in the 10-player Cleanup. Arrows indicate reward sharing.

conditional policy based on the mutual information constraint are shown in Figure 10(b). We conduct different randomized experiments in the 10-player Cleanup environment under 10 different random seeds and count the average extrinsic reward of the agents after the algorithm converges for each experiment. As can be seen from the figure, both the LToS+r algorithm and the D3C+r algorithm show significant and stable performance improvements.

### 4.5 The Impact of Rank Constraints

Finally, we perform an ablation analysis of rank constraints in RESVO. Intuitively, the number of roles, or the pattern of division of labor, will primarily affect the completion of tasks. The rank constraint $k$ represents a priori knowledge of the optimal number of roles in the task. In IPD, 3-player Escape Room, and Cleanup of different complexity, we set $k$, or the number of roles, to 1, 2, and 2, respectively. In this section, we want to verify the sensitivity of RESVO to the hyperparameter $k$. Similarly, we conduct 10 randomized experiments in the 10-player Cleanup environment for different $k$ and count the average external reward of the agents, and the results are shown in Figure 10(c).

It can be seen from the results in the figure that the RESVO algorithm is sensitive to the size of $k$. When the selection of $k$ is too large or too small, the performance will decrease significantly. The optimal number of roles in the 10-player Cleanup environment should be 2. However, it can be seen from the figure

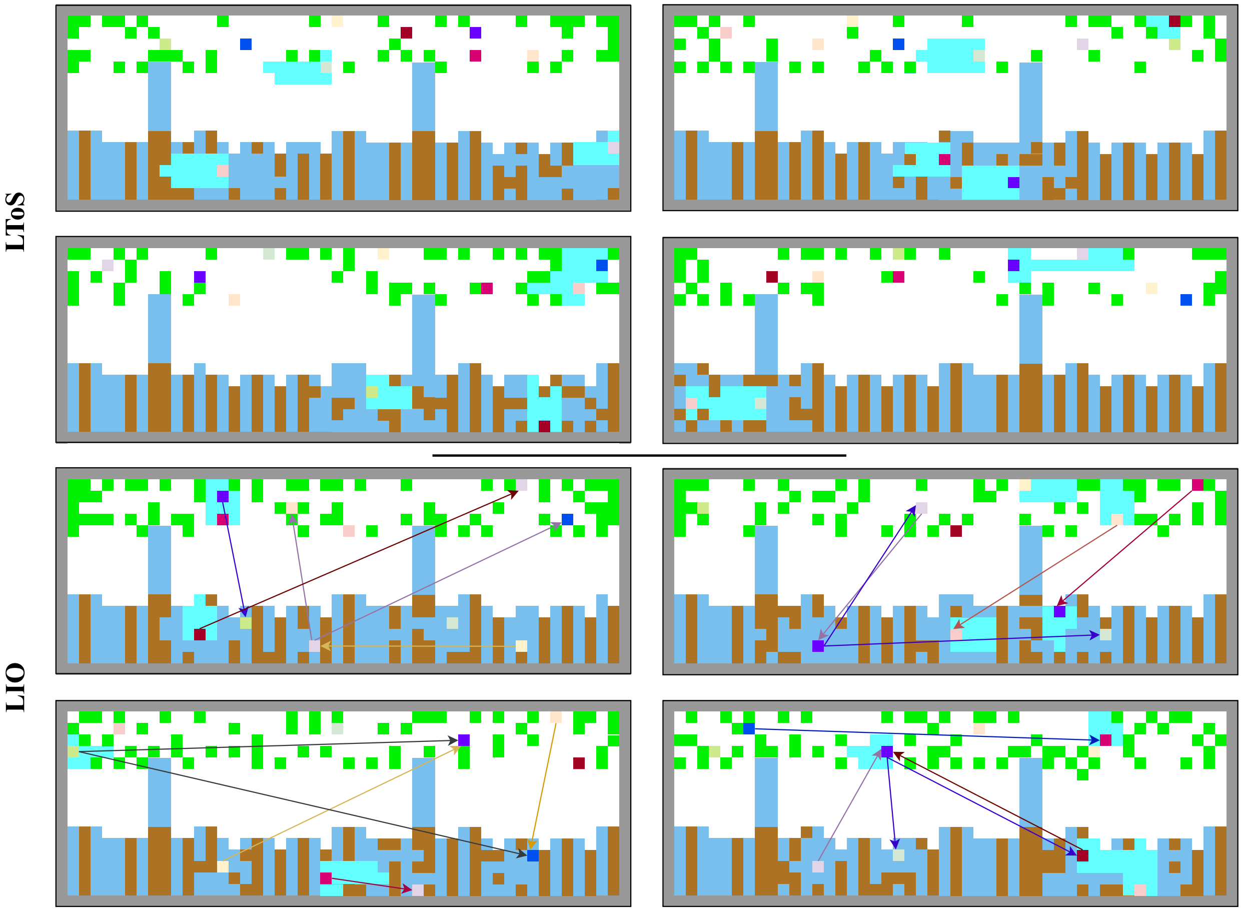


Figure 13: Selected keyframes from the rendered results of LToS and LIO in the 10-player Cleanup. Arrows indicate reward sharing.

that when $k$ is set to 3 or 4, the algorithm can also show promising results, indicating that the RESVO algorithm can show good robustness *near* the optimal $k$. We then present a more in-depth visual analysis of the SVO-based embeddings of emergent roles, which more intuitively shows the robustness of the RESVO algorithm around the optimal value of $k$.

Figure 14-17 show the SVO emergence in the training procedure of RESVO under different rank constraints in Cleanup with a map size of $48 \times 18$ and 10 agents. We visualize the SVO of each agent at the first timestep of a particular episode. We randomly map the SVO embedding of each agent to a $2D$ space using a fixed random neural network. This dimensionality reduction method can ensure that similar SVOs are also close together in $2D$ space. As seen from Figure 14, when the rank constraint is very low, all agents learn similar SVOs, that is, similar roles. This either means that all agents are free-riders or all agents have a composite role that needs to both collect apples and clean up waste. In either case, the overall performance will be poor.

When the size of the rank constraint is within a reasonable range, the algorithm can converge to the best performance, as shown in Figures 15 and 16. In previous results, the algorithm can converge to a better result when the rank constraint is near the optimal value ($k = 2$). Through the visualization results in Figures 15 and 16, we can propose an explanation experimentally. It can be seen from the two figures that although the agent is divided into more roles when $k = 4$, the similarity of these roles is different. Some roles are more similar, and others are less similar. Therefore, in the Cleanup, when $k = 4$, some two roles

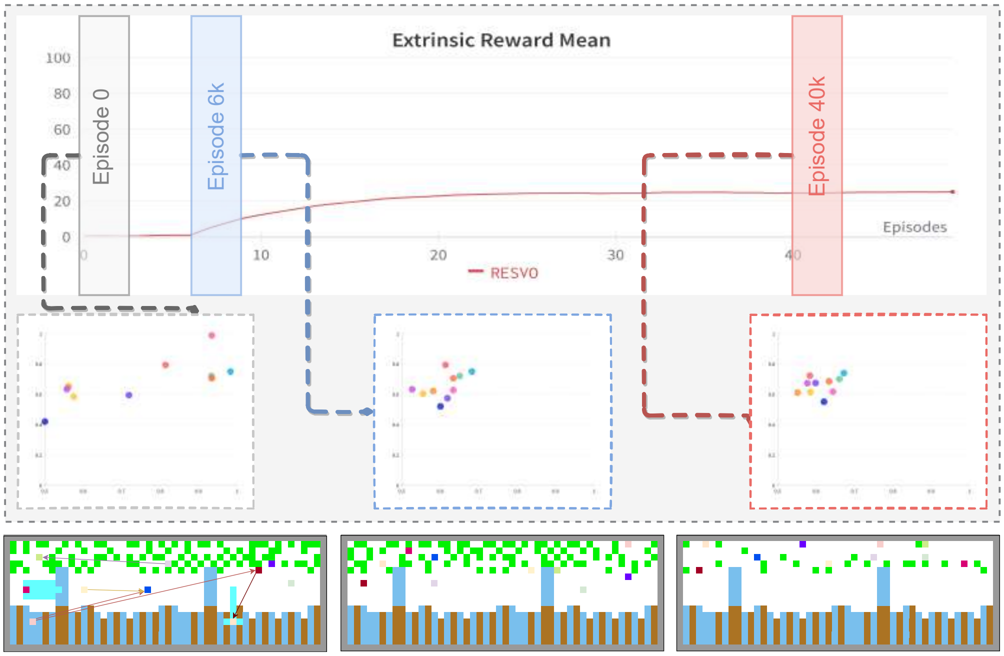


Figure 14: SVO-based role emergence with the rank $k = 1$ constrain.

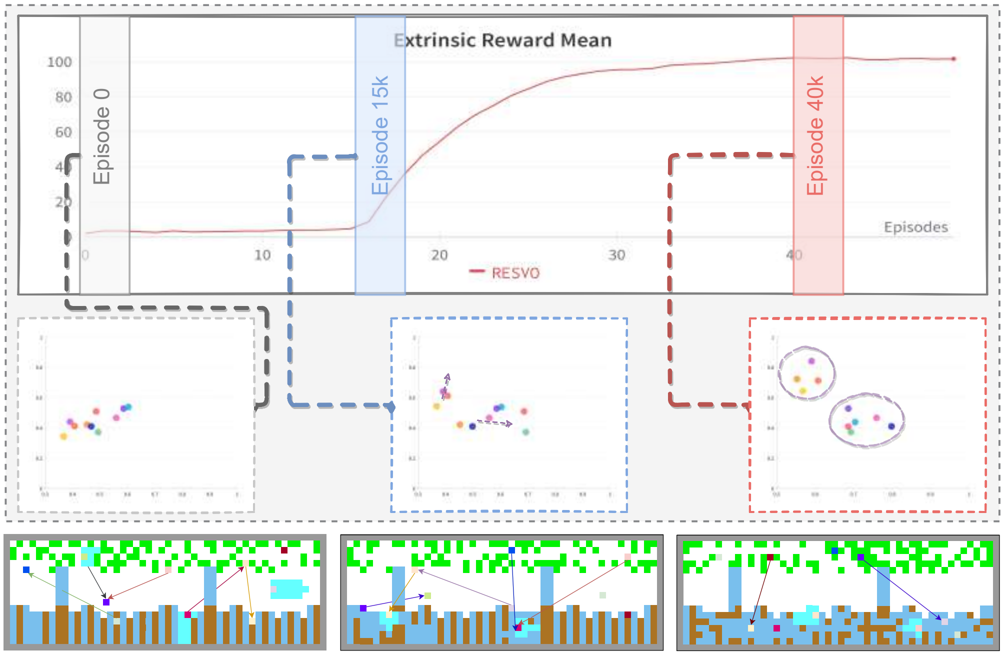


Figure 15: SVO-based role emergence with the rank $k = 2$ constrain.

may show a similar division of labor, so the performance will not be affected when there are more roles.

However, when the rank constraint increases ($k = 9$), the situation worsens. As shown in Figure 17, when the agent has too many roles, the SVO of the agent presents strong randomness, which makes the policy of the agent constrained by the SVO also present greater randomness. In the Cleanup task, the number of

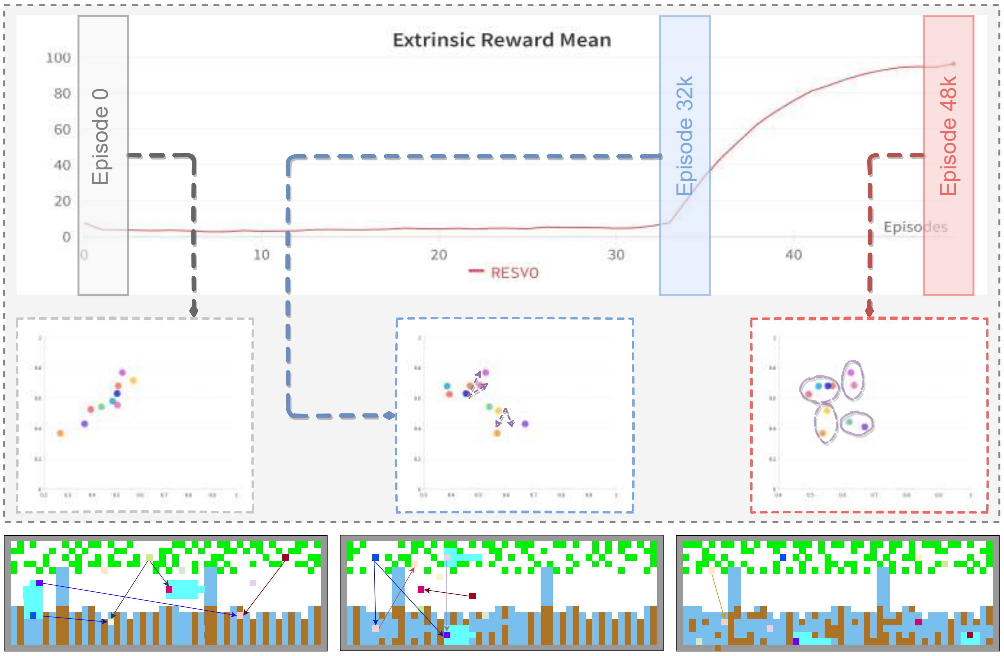


Figure 16: SVO-based role emergence with the rank $k = 4$ constrain.

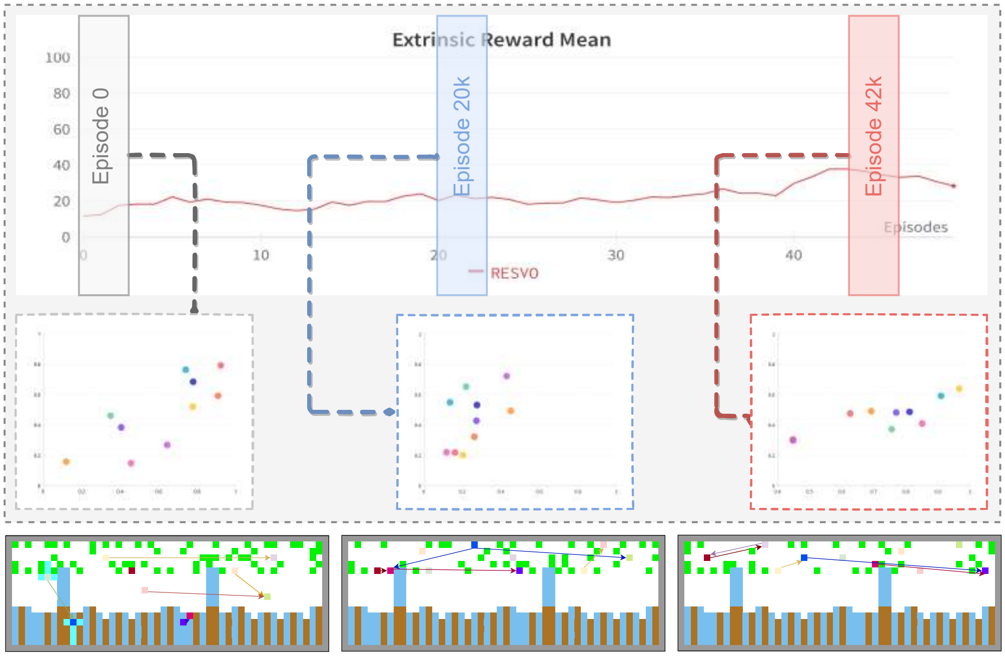


Figure 17: SVO-based role emergence with the rank $k = 9$ constrain.

agents that collect apples or clean up wastes will be small, and some agents will follow random policies and do useless work, which will not improve the algorithm's performance.

# 5 Closing Remarks

In this paper, we introduce a typical mechanism of human society, i.e., division of labor, to solve intertemporal social dilemmas (ISDs) in multi-agent reinforcement learning. A novel learning framework (RESVO) transforms role learning into a social value orientation emergence problem. RESVO solves it symmetrically by endowing agents with altruism to learn to share rewards with different weights with other agents. Numerical experiments on three tasks with different complexity in the presence of ISD show that RESVO can emerge stable roles and efficiently solve the ISD through the division of labor. Meanwhile, the SVO-based role emergence mechanism in RESVO can be introduced into other SVO-based methods as a plug-and-play module and bring a significant performance boost.

For the classic example, iterated prisoner's dilemma, we provide formal analyses and numerical results for the effect of social value orientation and division of labor in Appendix C. However, for more complex ISDs, i.e., $N$-Player Escape Room and Cleanup, we only provide empirical results in Section. 4. We aim to formally analyze the evolutionary dynamics of environmental behaviors, social value orientations, and roles of ISDs. However, the environmental policies, roles, and sharing policies are interdependent, and it is not easy to accurately account for the effect of the division of labor when considering environmental policy updates. These challenges have perplexed the researchers for a long time (Hirsch et al., 2012; Gould et al., 2016; Dong et al., 2021), and we believe that the solution to these questions is an essential and promising direction for future work.

In experiments, we find that the sanction-based LIO, the SVO-based LToS, D3C, and our RESVO take a completely different approach to maintain the division of labor. The former achieves a dynamic division of labor by continuously passing rewards among the agents, while the latter achieves a static division of labor by more sparse reward passing. In the IPD, 3-player Escape Room, and Cleanup tasks of varying complexity involved in the experiments, we find that static division of labor exhibits better performance compared to dynamic division of labor in tasks where the same role corresponds to multiple agents, such as the 3-player Escape Room, and the 10-player Cleanup, converging faster to better social welfare.

However, in this paper, we obtain the above conclusions by the social welfare of the algorithm only in a limited task. We believe the dynamic division of labor will be more advantageous than the static division of labor in specific tasks and certain evaluation metrics. For example, dynamic division of labor may be more robust in tasks that involve roles that change dynamically; furthermore, static division of labor may pose fairness issues because some agents receive lower extrinsic rewards than others. We leave the above questions for future exploration.

# References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P. A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *USENIX*.

Ahilan, S. and Dayan, P. (2021). Correcting experience replay for multi-agent communication. In *ICLR*.

Anastassacos, N., García, J., Hailes, S. and Musolesi, M. (2021). Cooperation and reputation dynamics with reinforcement learning. In *AAMAS*.

Anastassacos, N., Hailes, S. and Musolesi, M. (2020). Partner selection for the emergence of cooperation in multi-agent systems using reinforcement learning. In *AAAI*.

Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, **211** 1390–1396.

Baker, B. (2020). Emergent reciprocity and team formation from randomized uncertain social preferences. In *NeurIPS*.

Batson, C. D. (2012). A history of prosocial behavior research. In *Handbook of the history of social psychology*. 243–264.

Bonjean, N., Mefteh, W., Gleizes, M.-P., Maurel, C. and Migeon, F. (2014). Adelfe 2.0, handbook on agent-oriented design processes.

Bonnefon, J.-F., Shariff, A. and Rahwan, I. (2016). The social dilemma of autonomous vehicles. *Science*, **352** 1573–1576.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. and Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, **8** 203–236.

Butler, E. (2012). *The condensed wealth of nations*. Centre for Independent Studies.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A. and Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. In *NeurIPS*.

Cooper, D. J. and Kagel, J. H. (2016). Other-regarding preferences. In *The Handbook of Experimental Economics, Volume 2*. 217–289.

Cossentino, M., Gaglio, S., Sabatucci, L. and Seidita, V. (2005). The passi and agile passi mas meta-models compared with a unifying proposal. In *CEEMAS*.

Dafoe, A., Hughes, E., Bachrach, Y., Collins, T., McKee, K. R., Leibo, J. Z., Larson, K. and Graepel, T. (2020). Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*.

Davis, N. (2017). *The selfish gene*. Macat Library.

DeLoach, S. A. and Garcia-Ojeda, J. C. (2010). O-MaSE: a customisable approach to designing and building complex, adaptive multi-agent systems. *International Journal of Agent-Oriented Software Engineering*, **4** 244–280.

Dong, H., Wang, T., Liu, J., Han, C. and Zhang, C. (2021). Birds of a feather flock together: A close look at cooperation emergence via multi-agent rl. *arXiv preprint arXiv:2104.11455*.

Eccles, T., Hughes, E., Kramár, J., Wheelwright, S. and Leibo, J. Z. (2019). Learning reciprocity in complex sequential social dilemmas. *arXiv preprint arXiv:1903.08082*.

Eckel, C. C. and Grossman, P. J. (1996). Altruism in anonymous dictator games. *Games and economic behavior*, **16** 181–191.

Eisenberg, N. and Mussen, P. H. (1989). *The roots of prosocial behavior in children*. Cambridge University Press.

Encyclopædia Britannica (2022). Prisoners' dilemma. [Online; accessed Nov 2, 2022]. https://www.britannica.com/science/game-theory/images-videos#/media/1/224893/85154

Eysenbach, B., Gupta, A., Ibarz, J. and Levine, S. (2019). Diversity is all you need: Learning skills without a reward function. In *ICLR*.

Fehr, E. and Fischbacher, U. (2003). The nature of human altruism. *Nature*, **425** 785–791.

Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P. and Mordatch, I. (2018). Learning with opponent-learning awareness. In *AAMAS*.

Fullerton, D. and Metcalf, G. E. (2014). Environmental taxes and the double-dividend hypothesis: did you really expect something for nothing?

Gao, P., Hensley, R. and Zielke, A. (2014). A road map to the future for the auto industry. *McKinsey Quarterly, Oct* 1–11.

Gemp, I., McKee, K. R., Everett, R., Duéñez-Guzmán, E. A., Bachrach, Y., Balduzzi, D. and Tacchetti, A. (2022). D3c: Reducing the price of anarchy in multi-agent learning. In *AAMAS*.

Gigerenzer, G. and Selten, R. (2002). *Bounded rationality: The adaptive toolbox*. MIT press.

Gordon, D. M. (1996). The organization of work in social insect colonies. *Nature*, **380** 121–124.

Gould, S., Fernando, B., Cherian, A., Anderson, P., Cruz, R. S. and Guo, E. (2016). On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*.

Grimeaud, D.-E. (2001). An overview of the policy and legal aspects of the international climate change regime (part i). *Environmental Liability*, **9** 39–52.

Hansen, D. A. (1982). Interpersonal relations: A theory of interdependence.

Hansen, E. A., Bernstein, D. S. and Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *AAAI*.

Hirsch, M. W., Smale, S. and Devaney, R. L. (2012). *Differential equations, dynamical systems, and an introduction to chaos*. Academic press.

Holman, D. and Hughes, D. J. (2021). Transactions between big-5 personality traits and job characteristics across 20 years. *Journal of Occupational and Organizational Psychology*.

Hughes, E., Leibo, J. Z., Phillips, M., Tuyls, K., Dueñez-Guzman, E., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K., Koster, R. et al. (2018a). Inequity aversion improves cooperation in intertemporal social dilemmas. In *NeurIPS*.

Hughes, E., Leibo, J. Z., Phillips, M., Tuyls, K., Duéñez-Guzmán, E. A., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K. R., Koster, R., Roff, H. and Graepel, T. (2018b). Inequity aversion improves cooperation in intertemporal social dilemmas. In *NeurIPS*.

Ibrahim, A., Jitani, A., Piracha, D. and Precup, D. (2020a). Reward redistribution mechanisms in multi-agent reinforcement learning. In *AAMAS Workshop on Adaptive and Learning Agents*.

Ibrahim, A., Jitani, A., Piracha, D. and Precup, D. (2020b). Reward redistribution mechanisms in multi-agent reinforcement learning. In *AAMAS Adaptive and Learning Agents Workshop*.

Institute, A. S. (2013). On the wealth of nations.

Ivanov, D., Egorov, V. and Shpilman, A. (2021). Balancing rational and other-regarding preferences in cooperative-competitive environments. In *AAMAS*.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A. et al. (2019). Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, **364** 859–865.

Jeanson, R., Kukuk, P. F. and Fewell, J. H. (2005). Emergence of division of labour in halictine bees: contributions of social interactions and behavioural variance. *Animal Behaviour*, **70** 1183–1193.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K. A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H. and Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, **14** 1–210.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*.

Kollock, P. (1998). Social dilemmas: The anatomy of cooperation. *Annual Review of Sociology*, **24** 183–214.

Koster, R., Hadfield-Menell, D., Hadfield, G. K. and Leibo, J. Z. (2020). Silly rules improve the capacity of agents to learn stable enforcement and compliance behaviors. In *AAMAS*.

Köster, R., McKee, K. R., Everett, R., Weidinger, L., Isaac, W. S., Hughes, E., Duéñez-Guzmán, E. A., Graepel, T., Botvinick, M. and Leibo, J. Z. (2020). Model-free conventions in multi-agent reinforcement learning with heterogeneous preferences. *arXiv preprint arXiv:2010.09054*.

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J. and Graepel, T. (2017). Multi-agent reinforcement learning in sequential social dilemmas. In *AAMAS*.

Lhaksmana, K. M., Murakami, Y. and Ishida, T. (2018). Role-based modeling for designing agent behavior in self-organizing multi-agent systems. *International Journal of Software Engineering and Knowledge Engineering*, **28** 79–96.

Li, W. (2022). *Cooperation Promotion Multi-Agent Reinforcement Learning*. Ph.D. thesis, East China Normal University.

Li, W., Wang, X., Jin, B., Luo, D. and Zha, H. (2022). Structured cooperative reinforcement learning with time-varying composite action space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44** 8618–8634.

Liao, X., Li, W., Xu, Q., Wang, X., Jin, B., Zhang, X., Zhang, Y. and Wang, Y. (2020). Iteratively-refined interactive 3d medical image segmentation with multi-agent reinforcement learning. In *CVPR*.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P. and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*.

Lupu, A. and Precup, D. (2020). Gifting in multi-agent reinforcement learning. In *AAMAS*.

Majeski, S. J. (1984). Arms races as iterated prisoner's dilemma games. *Mathematical Social Sciences*, **7** 253–266.

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. In *AAMAS*.

Omicini, A. (2000). Soda: Societies and infrastructures in the analysis and design of agent-based systems. In *International Workshop on AOSE*.

Orbell, J., Dawes, R. and Van de Kragt, A. (1990). The limits of multilateral promising. *Ethics*, **100** 616–627.

Orbell, J. M., Van de Kragt, A. J. and Dawes, R. M. (1988). Explaining discussion-induced cooperation. *Journal of Personality and Social Psychology*, **54** 811.

Ostrom, E. (1990). *Governing the commons: The evolution of institutions for collective action*. Cambridge university press.

Padgham, L. and Winikoff, M. (2002). Prometheus: A methodology for developing intelligent agents. In *International Workshop on AOSE*.

Pavón, J. and Gómez-Sanz, J. (2003). Agent oriented software engineering with INGENIAS. In *CEEMAS*.

Pennisi, E. (2009). On the origin of cooperation. *Science*, **325** 1196–1199.

Peysakhovich, A. and Lerer, A. (2018a). Consequentialist conditional cooperation in social dilemmas with imperfect information. In *ICLR*.

Peysakhovich, A. and Lerer, A. (2018b). Prosocial learning agents solve generalized stag hunts better than selfish ones. In *AAMAS*.

Pretorius, A., Cameron, S., Van Biljon, E., Makkink, T., Mawjee, S., du Plessis, J., Shock, J., Laterre, A. and Beguir, K. (2020). A game-theoretic analysis of networked system control for common-pool resource management using multi-agent reinforcement learning. In *NeurIPS*.

Rapoport, A., Chammah, A. M. and Orwant, C. J. (1965). *Prisoner's dilemma: A study in conflict and cooperation*, vol. 165. University of Michigan Press.

Rehmeyer, J. (2012). Game theory suggests current climate negotiations won't avert catastrophe. *Science News*.

Rushton, J. P., Chrisjohn, R. D. and Fekken, G. C. (1981). The altruistic personality and the self-report altruism scale. *Personality and individual differences*, **2** 293–302.

Santos, F. P., Pacheco, J. M. and Santos, F. C. (2021). The complexity of human cooperation under indirect reciprocity. *Philosophical Transactions of the Royal Society B*, **376** 20200291.

Schmid, K., Belzner, L., Müller, R., Tochtermann, J. and Linnhoff-Popien, C. (2021). Stochastic market games. In *IJCAI*.

Schneier, B. (2012). Lance armstrong and the prisoners' dilemma of doping in professional sports, wired.

Schoemaker, P. J. (2013). *Experiments on decisions under risk: The expected utility hypothesis*. Springer Science & Business Media.

Sheng, J., Wang, X., Jin, B., Yan, J., Li, W., Chang, T.-H., Wang, J. and Zha, H. (2020). Learning structured communication for multi-agent reinforcement learning. *arXiv preprint arXiv:2002.04235*.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T. P., Simonyan, K. and Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, **362** 1140–1144.

Silver, D., Singh, S., Precup, D. and Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, **299** 103535.

Simon, H. A. (1993). Altruism and economics. *The American Economic Review*, **83** 156–161.

Spanoudakis, N. and Moraitis, P. (2010). Using ASEME methodology for model-driven agent systems development. In *International Workshop on AOSE*.

Spieser, K., Treleaven, K., Zhang, R., Frazzoli, E., Morton, D. and Pavone, M. (2014). Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore. In *Road vehicle automation*. 229–245.

Sun, C., Liu, W. and Dong, L. (2020). Reinforcement learning with task decomposition for cooperative multiagent systems. *IEEE Transactions on Neural Networks and Learning Systems*, **32** 2054–2065.

Sutin, A. R. and Costa, P. T. (2010). Reciprocal influences of personality and job characteristics across middle adulthood. *Journal of Personality*, **78** 257–88.

Sutton, R. S., McAllester, D. A., Singh, S. and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*.

Tuyls, K., Perolat, J., Lanctot, M., Leibo, J. and Graepel, T. (2018). A generalised method for empirical game theoretic analysis. In *AAMAS*.

Vamplew, P., Smith, B. J., Källström, J., Ramos, G., Rădulescu, R., Roijers, D. M., Hayes, C. F., Heintz, F., Mannion, P., Libin, P. J. et al. (2022). Scalar reward is not enough: A response to silver, singh, precup and sutton (2021). *Autonomous Agents and Multi-Agent Systems*, **36** 1–19.

Van Arem, B., Van Driel, C. J. and Visser, R. (2006). The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on intelligent transportation systems*, **7** 429–436.

Vinitsky, E., Jaques, N., Leibo, J., Castenada, A. and Hughes, E. (2019). An open source implementation of sequential social dilemma games.

Vinitsky, E., Köster, R., Agapiou, J. P., Duéñez-Guzmán, E., Vezhnevets, A. S. and Leibo, J. Z. (2021). A learning agent that acquires social norms from public sanctions in decentralized multi-agent settings. *arXiv preprint arXiv:2106.09012*.

von Neumann, J. and Morgenstern, O. (2007). Theory of games and economic behavior (commemorative edition, 60th-anniversary edition). *With an introduction by Harold Kuhn and Ariel Rubinstein*.

Wang, J. X., Hughes, E., Fernando, C., Czarnecki, W. M., Duéñez-Guzmán, E. A. and Leibo, J. Z. (2019). Evolving intrinsic motivations for altruistic behavior. In *AAMAS*.

Wang, T., Dong, H., Lesser, V. R. and Zhang, C. (2020). Roma: Multi-agent reinforcement learning with emergent roles. In *ICML*.

Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S. and Zhang, C. (2021). Rode: Learning roles to decompose multi-agent tasks. In *ICLR*.

Wilson, A., Fern, A. and Tadepalli, P. (2010). Bayesian policy search for multi-agent role discovery. In *AAAI*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. et al. (2020). Transformers: State-of-the-art natural language processing. In *EMNLP*.

Wooldridge, M., Jennings, N. R. and Kinny, D. (2000). The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-agent Systems*, **3** 285–312.

Xu, J., Garcia, J. and Handfield, T. (2019). Cooperation with bottom-up reputation dynamics. In *AAMAS*.

Yang, J., Li, A., Farajtabar, M., Sunehag, P., Hughes, E. and Zha, H. (2020). Learning to incentivize other learning agents. In *NeurIPS*.

Yang, Y. (2021). *Many-agent Reinforcement Learning*. Ph.D. thesis, UCL (University College London).

Yi, Y., Li, G., Wang, Y. and Lu, Z. (2021). Learning to share in multi-agent reinforcement learning. *arXiv preprint arXiv:2112.08702*.

Zhang, C., Li, X., Hao, J., Chen, S., Tuyls, K., Xue, W. and Feng, Z. (2019). Sa-iga: a multiagent reinforcement learning method towards socially optimal outcomes. *Autonomous Agents and Multi-Agent Systems*, **33** 403–429.

Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C. and Socher, R. (2022). The ai economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, **8** eabk2607.

Zhu, H. and Zhou, M. (2008). Role-based multi-agent systems. In *Personalized Information Retrieval and Access: Concepts, Methods and Practices.* 254–285.

# A Implementation Details

Our method is built on the open-sourced codebase LIO (Yang et al., 2020), ROMA (Wang et al., 2020) and Sequential Social Dilemma Games (SSDG) (Vinitsky et al., 2019). We test our method on two Cleanup maps with different numbers of agents (i.e., 2 and 10), and the details of each map are shown in Table 1.

Table 1: Environmental settings for Cleanup with different numbers of agents.

| Parameter | n=2 | n=10 |
|---|---|---|
| Map Size | 10 × 10 | 48 × 18 |
| View Size | 7 | 15 |
| Max Steps | 50 | 400 |
| Apple Respawn Probability | 0.3 | 0.15 |
| Depletion Threshold | 0.4 | 0.99 |
| Restoration Threshold | 0.0 | 0.0 |
| Waste Spawn Probability | 0.5 | 0.15 |

We use fully-connected neural and convolutional networks for function approximation in the [IPD, ER] and Cleanup, respectively. All algorithms use the same neural architecture without sharing parameters within each intertemporal social dilemma. The output of each agent's orientation function is scaled element-wise by a multiplier $w_{\max}$, which is bounded in $[-w_{\max}, w_{\max}]$. The multiplier $w_{\max}$ is set to $[3, 2, 2]$ for [IPD, ER, Cleanup], respectively. We use a policy gradient algorithm to train the policy in IPD, ER, and actor-critic for Cleanup. An exploration lower bound $\epsilon$ is used to perform sufficient exploration, with $\epsilon$ decaying linearly from $\epsilon_{\text{start}}$ to $\epsilon_{\text{end}}$ by $\epsilon_{\text{div}}$ episodes. The values of these hyperparameters in different tasks are consistent with LIO (Yang et al., 2020). We use discount factor $\gamma = 0.99$, the gradient descent for policy optimization, and the Adam (Kingma and Ba, 2015) optimizer for training value and orientation functions.

For the trajectory encoder, we build the causal transformer (Chen et al., 2021) implementation for Cleanup off of minGPT[8], a publicly available reimplementation of GPT. We use most of the hyperparameters from Decision Transformer Chen et al. (2021)[9]. Moreover, for Iterated Prisoner's Dilemma and $N$-Player Escape Room, our code is based on the Huggingface Transformers library (Wolf et al., 2020). Our hyperparameters on these tasks are also the same with Chen et al. (2021).

We compare our method against various baselines. For ROMA (Wang et al., 2020), LIO (Yang et al., 2020), LToS (Yi et al., 2021), and D3C (Gemp et al., 2022), we use the codes provided by the authors and the hyper-parameters that have been fine-tuned on the Sequential Social Dilemma Games (SSDG) (Vinitsky et al., 2019). The ablations LToS+r and D3C+r use the exact implementation and hyperparameter settings as LToS and D3C, respectively, but with the only difference that the role emergence mechanism is introduced, which is the same as in our method. Learning curves are smoothed by averaging over a window of 11 episodes. The hardware used in the experiment is a server with 128G memory and 4 NVIDIA 1080Ti graphics cards with 11G video memory. The code and license of baselines are shown in the following list:

- ROMA (Wang et al., 2020): https://github.com/TonghanWang/ROMA, Apache-2.0 License;
- LIO (Yang et al., 2020): https://github.com/011235813/lio, MIT License;
- SSDG (Vinitsky et al., 2019): https://github.com/eugenevinitsky/sequential_social_dilemma_games, MIT License.

[8] https://github.com/karpathy/minGPT.

[9] https://github.com/kzl/decision-transformer, MIT License.

# B Derivations

## B.1 Role Emergence

Here we deduce the (4) in detail. For logical integrity, we have merged and reorganized some of the contents of Section 3.1 and Section 3.2. At each time step $t$, each agent $j$ receives a total reward

$$r_j(\boldsymbol{\eta}, \boldsymbol{r}) := w_{\eta_j}^j[j] \cdot r_j + \sum_{i \neq j} w_{\eta_i}^i[j] \cdot r_i, \tag{9}$$

Each agent $j$ learns a SVO-based role conditioned policy $\pi_j(\cdot \mid o_j, e_j(\boldsymbol{\eta}))$ parameterized by $\theta_j$, where $e_j(\cdot)$ is the SVO-based role embedding, to maximize the objective

$$\max_{\theta_j} J^{\text{policy}}(\theta_j, \{e_j\}) := \mathbb{E}_{\boldsymbol{\pi}(\cdot|\cdot,\{e_j\})}\left[\sum_{t=0}^{T} \gamma^t r_j^t(\boldsymbol{\eta}, \boldsymbol{r})\right]. \tag{10}$$

Upon experiencing trajectories $\boldsymbol{\tau} := \{(s^0, \mathbf{a}^0, \mathbf{r}^0, \ldots, s^T)\}$, the recipient carries out an update

$$\hat{\theta}_j \leftarrow \theta_j + \beta f\left(\boldsymbol{\tau}, \theta_j, \eta\right). \tag{11}$$

that adjusts its policy parameters with learning rate $\beta$. Assuming policy optimization learners in this work and choosing policy gradient for exposition, the update function is

$$f\left(\boldsymbol{\tau}, \theta_j, \eta\right) = \sum_{t=0}^{T} \nabla_{\theta_j} \log \pi_j\left(a_j^t \mid o_j^t, e_j^t(\boldsymbol{\eta})\right) G_j^t\left(\boldsymbol{\tau}; \boldsymbol{\eta}\right), \tag{12}$$

where the return $G_j^t\left(\boldsymbol{\tau}, \boldsymbol{\eta}\right) = \sum_{l=t}^{T} \gamma^{l-t} r_j\left(\boldsymbol{\eta}, \mathbf{r}_l\right)$ depends on orientation parameters $\boldsymbol{\eta}$. After each agent has updated its policy to $\hat{\pi}_j$, parameterized by new $\hat{\theta}_j$, it generates a new trajectory $\hat{\tau}_j$. Using these trajectories, each sharing agent $i$ updates shared orientation function parameters $\eta_i$ to maximize the following individual objective:

$$\max_{\eta_i} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) := \mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\sum_{t=0}^{T} \gamma^t \left(\hat{r}_i^t - \alpha \|W_{\eta_i}^{i,t} - W_k^{i,t}\|_2^2\right)\right], \tag{13}$$

where the superscription $i$ denotes the $i$-th column of the matrix $W_{\eta_i}^{i,t}$. The penalty term introduced by the rank constraint can be regarded as an intrinsic reward $-\alpha\|W_\eta^{i,t} - W_k^{i,t}\|_2^2$. And the gradient of $J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right)$ w.r.t. $\eta_i$ is:

$$\begin{aligned}
&\nabla_{\eta_i} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) \\
=&\nabla_{\eta_i} \hat{\boldsymbol{\theta}} \nabla_{\hat{\boldsymbol{\theta}}} J_i^{\text{svo}}\left(\hat{\tau}^i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) + \nabla_{\eta_i} \boldsymbol{\eta} \nabla_{\boldsymbol{\eta}} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) + \nabla_{\eta_i} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) \\
=&\underbrace{\sum_{j=1}^{N} (\nabla_{\eta_i} \hat{\theta}_j)^T \nabla_{\hat{\theta}_j} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right)}_{\text{blue part}} + \underbrace{(\nabla_{\eta_i} \boldsymbol{\eta})^T \nabla_{\boldsymbol{\eta}} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right)}_{\text{orange part}} + \\
&\underbrace{\nabla_{\eta_i} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right)}_{\text{purple part}}.
\end{aligned} \tag{14}$$

The first factor of each term in the summation of the blue part follows directly from (11) and (12):

$$\begin{aligned}
\nabla_{\eta_i} \hat{\theta}_j = \beta \sum_{t=0}^{T} &\nabla_{\eta_i} \nabla_{\theta_j} \log \pi_j\left(a_j^t \mid o_j^t, e_j(\boldsymbol{\eta})\right) G_j^t\left(\boldsymbol{\tau}; \boldsymbol{\eta}\right) + \\
&\nabla_{\theta_j} \log \pi_j\left(a_j^t \mid o_j^t, e_j(\boldsymbol{\eta})\right) \left(\nabla_{\eta_i} G_j^t\left(\boldsymbol{\tau}; \boldsymbol{\eta}\right)\right)^T.
\end{aligned} \tag{15}$$

Note that (11) does not contain recursive dependence of $\theta_j$ on $\eta$ since $\theta_j$ is a function of incentives in *previous* episodes and orientation parameters, not those in trajectory $\tau_i$ and current $\eta$. The second factor of the blue part can be derived similarly as standard policy gradients (Sutton et al., 1999):

$$\begin{aligned}
&\nabla_{\hat{\theta}_j} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) = \nabla_{\hat{\theta}^j} \tilde{V}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \eta_i\right) = \nabla_{\hat{\theta}^j} \sum_{\hat{\mathbf{a}}} \hat{\boldsymbol{\pi}}\left(\hat{\mathbf{a}} \mid \hat{s}_0, \hat{\boldsymbol{e}}(\boldsymbol{\eta})\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) \\
=&\sum_{\hat{\mathbf{a}}} \hat{\pi}_{\hat{\theta}_{-j}}^{-j}\left(\left(\nabla_{\hat{\theta}_j} \hat{\pi}_{\hat{\theta}_j}^j\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) + \hat{\pi}_{\hat{\theta}_j}^j \nabla_{\hat{\theta}_j} \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right)\right) \\
=&\sum_{\hat{\mathbf{a}}} \hat{\pi}_{\hat{\theta}_{-j}}^{-j}\left(\left(\nabla_{\hat{\theta}_j} \hat{\pi}_{\hat{\theta}_j}^j\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) + \hat{\pi}_{\hat{\theta}_j}^j \nabla_{\hat{\theta}_j}\left(\hat{r}_i + \gamma \sum_{\hat{s}'} P\left(\hat{s}' \mid \hat{s}_0, \hat{\mathbf{a}}\right) \tilde{V}_i^{\boldsymbol{\pi}}\left(\hat{s}', \eta_i\right)\right)\right) \\
=&\sum_{\hat{\mathbf{a}}} \hat{\pi}_{\hat{\theta}_{-j}}^{-j}\left(\left(\nabla_{\hat{\theta}_j} \hat{\pi}_{\hat{\theta}_j}^j\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) + \gamma \hat{\pi}_{\hat{\theta}_j}^j \sum_{\hat{s}'} P\left(\hat{s}' \mid \hat{s}_0, \hat{\mathbf{a}}\right) \nabla_{\hat{\theta}_j} \tilde{V}_i^{\boldsymbol{\pi}}\left(\hat{s}', \eta_i\right)\right) \\
=&\sum_x \sum_{k=0}^T P\left(\hat{s}_0 \rightarrow x, k, \hat{\boldsymbol{\pi}}\right) \gamma^k \sum_{\hat{\mathbf{a}}} \hat{\pi}_{\hat{\theta}_{-j}}^{-j} \nabla_{\hat{\theta}_j} \hat{\pi}_{\hat{\theta}_j}^j \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(x, \hat{\mathbf{a}}, \eta_i) \\
=&\sum_{\hat{s}} d^{\hat{\boldsymbol{\pi}}}(\hat{s}) \sum_{\hat{\mathbf{a}}} \hat{\pi}_{\hat{\theta}_{-j}}^{-j} \nabla_{\hat{\theta}_j} \hat{\pi}_{\hat{\theta}_j}^j \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(\hat{s}, \hat{\mathbf{a}}, \eta_i) \\
=&\sum_{\hat{s}} d^{\hat{\boldsymbol{\pi}}}(\hat{s}) \sum_{\hat{\mathbf{a}}} \hat{\pi}_{\hat{\theta}_{-j}}^{-j} \hat{\pi}_{\hat{\theta}_j}^j \nabla_{\hat{\theta}_j} \log \hat{\pi}_{\hat{\theta}_j}^j \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(\hat{s}, \hat{\mathbf{a}}, \eta_i) \\
=&\mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\nabla_{\hat{\theta}^j} \log \hat{\pi}_{\hat{\theta}_j}^j\left(\hat{a}_j \mid \hat{o}_j, \hat{e}_j(\boldsymbol{\eta})\right) \tilde{Q}^{i, \hat{\boldsymbol{\pi}}}(\hat{s}, \hat{\mathbf{a}}, \eta_i)\right],
\end{aligned} \tag{16}$$

where $\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(\hat{s}, \hat{\mathbf{a}}, \boldsymbol{\eta})$ represents the $Q$-function of the augmented reward which is augmented by the intrinsic reward $-\alpha\|W_{\eta_i}^{i,t} - W_k^{i,t}\|_2^2$, $\hat{\pi}_{\hat{\theta}_{-j}}^{-j} := \hat{\pi}_{\hat{\theta}_{-j}}^{-j}\left(\hat{a}_{-j} \mid \hat{s}_0, \hat{e}_{-j}(\boldsymbol{\eta})\right)$, $\hat{\pi}_{\hat{\theta}_j}^j := \hat{\pi}_{\hat{\theta}_j}^j\left(\hat{a}_j \mid \hat{s}_0, \hat{e}_j(\boldsymbol{\eta})\right)$. Similar to the derivation of the second factor of the blue part, the second factor of the orange part can be derived as

$$\begin{aligned}
&\nabla_{\boldsymbol{\eta}} J_i^{\text{svo}}\left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) = \nabla_{\boldsymbol{\eta}} \tilde{V}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \eta_i\right) = \nabla_{\boldsymbol{\eta}} \sum_{\hat{\mathbf{a}}} \hat{\boldsymbol{\pi}}\left(\hat{\mathbf{a}} \mid \hat{s}_0, \hat{\boldsymbol{e}}(\boldsymbol{\eta})\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) \\
=&\sum_{\hat{\mathbf{a}}}\left(\left(\nabla_{\boldsymbol{\eta}} \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) + \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\eta}} \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right)\right) \\
=&\sum_{\hat{\mathbf{a}}}\left(\left(\nabla_{\boldsymbol{\eta}} \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) + \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\eta}}\left(\hat{r}_i + \gamma \sum_{\hat{s}'} P\left(\hat{s}' \mid \hat{s}_0, \hat{\mathbf{a}}\right) \tilde{V}_i^{\boldsymbol{\pi}}\left(\hat{s}', \eta_i\right)\right)\right) \\
=&\sum_{\hat{\mathbf{a}}}\left(\left(\nabla_{\boldsymbol{\eta}} \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\right) \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0, \hat{\mathbf{a}}, \eta_i\right) + \gamma \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}} \sum_{\hat{s}'} P\left(\hat{s}' \mid \hat{s}_0, \hat{\mathbf{a}}\right) \nabla_{\boldsymbol{\eta}} \tilde{V}_i^{\boldsymbol{\pi}}\left(\hat{s}', \eta_i\right)\right) \\
=&\sum_x \sum_{k=0}^T P\left(\hat{s}_0 \rightarrow x, k, \hat{\boldsymbol{\pi}}\right) \gamma^k \sum_{\hat{\mathbf{a}}} \nabla_{\boldsymbol{\eta}} \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}} \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(x, \hat{\mathbf{a}}, \eta_i) \\
=&\sum_{\hat{s}} d^{\hat{\boldsymbol{\pi}}}(\hat{s}) \sum_{\hat{\mathbf{a}}} \nabla_{\boldsymbol{\eta}} \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}} \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(\hat{s}, \hat{\mathbf{a}}, \eta_i) \\
=&\sum_{\hat{s}} d^{\hat{\boldsymbol{\pi}}}(\hat{s}) \sum_{\hat{\mathbf{a}}} \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\eta}} \log \hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}} \tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(\hat{s}, \hat{\mathbf{a}}, \eta_i) \\
=&\mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\left(\sum_{j=1}^N \nabla_{\boldsymbol{\eta}} \log \hat{\pi}_{\hat{\theta}_j}^j\left(\hat{a}_j \mid \hat{o}_j, \hat{e}_j(\boldsymbol{\eta})\right)\right) \tilde{Q}^{i, \hat{\boldsymbol{\pi}}}(\hat{s}, \hat{\mathbf{a}}, \eta_i)\right].
\end{aligned} \tag{17}$$

Finally, the purple part can be directly derived as

$$\nabla_{\eta_i} J_i^{\text{svo}} \left(\hat{\tau}_i, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}, \eta_i\right) = \mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\sum_{t=0}^{T} -2\alpha\gamma^t \nabla_{\eta_i} W_{\eta_i}^{i,t}\left(W_{\eta_i}^{i,t} - W_k^{i,t}\right)\right]. \tag{18}$$

## B.2 Policy Optimization

Introducing SVO-based role embedding and conditioning individual policies on this embedding explicitly establish the connection between the role and the individual policies to encourage the division of labor through the diversity of roles. Nevertheless, this does not enable the role representation to exhibit sufficient responsibilities. That is, the role can constrain the long-term behavior of the agent. Drawing inspiration from Eysenbach et al. (2019); Wang et al. (2020), we propose to learn SVO-based roles that are identifiable by agents' long-term behaviors, which can be achieved by maximizing $I(\tau_i; e_i \mid \boldsymbol{o}, \boldsymbol{a})$, the conditional mutual information between the individual trajectory and the role given the current *joint* observation and *joint* action.

Based on variational inference and the derivation process in Wang et al. (2020), a variational posterior estimator can be proposed to derive a tractable lower bound for the mutual information objective

$$\begin{aligned}
& I\left(e_i^t; \tau_i^{t-1} \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right) \\
& = \mathbb{E}_{e_i^t, \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log \frac{p\left(\rho_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)}{p\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)}\right] \\
& = \mathbb{E}_{e_i^t, \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log \frac{q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)}{p\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)}\right] \\
& \quad + \mathbb{E}_{\tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[D_{\mathrm{KL}}\left(p\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right) \| q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right)\right] \\
& \geq \mathbb{E}_{e_i^t, \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log \frac{q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)}{p\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)}\right],
\end{aligned} \tag{19}$$

where the last inequality holds via non-negativity of the KL divergence. Then it follows that:

$$\begin{aligned}
& \mathbb{E}_{e_i^t, \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log \frac{q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)}{p\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)}\right] \\
= & \mathbb{E}_{e_i^t, \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right] - \mathbb{E}_{e_i^t, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log p\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right] \\
= & \mathbb{E}_{e_i^t, \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\log q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right] + \mathbb{E}_{\boldsymbol{o}^t, \boldsymbol{a}^t}\left[H\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right] \\
= & \mathbb{E}_{\tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\int p\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right) \log q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right) de_i^t\right] \\
& \quad + \mathbb{E}_{\boldsymbol{o}^t, \boldsymbol{a}^t}\left[H\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right].
\end{aligned} \tag{20}$$

The role encoder is conditioned on the joint observations and actions, so given the observations and actions, the distributions of roles, $p\left(e_i^t\right)$, are independent from the local histories. Thus, we have

$$\begin{aligned}
I\left(e_i^t; \tau_i^{t-1} \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right) \geq & -\mathbb{E}_{\tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\mathcal{CE}\left[p\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right) \| q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right]\right] \\
& + \mathbb{E}_{\boldsymbol{o}^t, \boldsymbol{a}^t}\left[H\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right].
\end{aligned} \tag{21}$$

In SVO-based role emergence, we use orientation functions to act as the role encoder. Therefore, the above formula can be transformed into

$$\begin{aligned}
I\left(e_i^t; \tau_i^{t-1} \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right) \geq & -\mathbb{E}_{\tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t}\left[\mathcal{CE}\left[W_{\boldsymbol{\eta}}\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right) \| q_\phi\left(e_i^t \mid \tau_i^{t-1}, \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right]\right] \\
& + \mathbb{E}_{\boldsymbol{o}^t, \boldsymbol{a}^t}\left[H_{\boldsymbol{\eta}}\left(e_i^t \mid \boldsymbol{o}^t, \boldsymbol{a}^t\right)\right].
\end{aligned} \tag{22}$$

In practice, we use a replay buffer $\mathcal{D}$ and minimize the following summation of all agents

$$\begin{aligned} &\mathrm{L}^{\mathrm{mi}}\left(\boldsymbol{\tau};\boldsymbol{\eta},\phi\right)=\frac{1}{n}\sum_{i=1}^{N}\mathrm{L}_i^{\mathrm{mi}}\left(\tau_i;\boldsymbol{\eta},\phi\right) \\ :=&\mathbb{E}_{\left(\tau_i^{t-1},\boldsymbol{o}^t,\boldsymbol{a}^t\right)\sim\mathcal{D}}\left[\mathcal{CE}\left[W_{\boldsymbol{\eta}}\left(e_i^t\mid\boldsymbol{o}^t,\boldsymbol{a}^t\right)\|q_\phi\left(e_i^t\mid\tau_i^{t-1},\boldsymbol{o}^t,\boldsymbol{a}^t\right)\right]-H_{\boldsymbol{\eta}}\left(e_i^t\mid\boldsymbol{o}^t,\boldsymbol{a}^t\right)\right] \\ =&\mathbb{E}\left[D_{\mathrm{KL}}\left[W_{\boldsymbol{\eta}}\left(e_i^t\mid\boldsymbol{o}^t,\boldsymbol{a}^t\right)\|q_\phi\left(e_i^t\mid\tau_i^{t-1},\boldsymbol{o}^t,\boldsymbol{a}^t\right)\right]\right], \end{aligned} \tag{23}$$

In addition, in the implementation, the orientation functions model $p$ as a normal distribution and output the mean and variance of the SVO-based role embedding, which can be sampled from the corresponding normal distribution. However, in the training process of role emergence, we only use the mean of the output of orientation functions.

## B.3 Practice Implementation

From the derivation of (14), it can be seen that the gradient calculation involves the calculation of multi-part partial derivatives and complex matrix-vector multiplication. Alternatively, one may rely on automatic differentiation in modern machine learning frameworks (Abadi et al., 2016) to compute the chain rule (14) via direct minimization of a loss function. This loss function is derived as follows:

$$\begin{aligned} &\nabla_{\eta_i}J_i^{\mathrm{svo}}\left(\hat{\tau}_i,\hat{\boldsymbol{\theta}},\boldsymbol{\eta},\eta_i\right)=\nabla_{\eta_i}\tilde{V}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0,\eta_i\right)=\nabla_{\eta_i}\sum_{\hat{\mathbf{a}}}\hat{\boldsymbol{\pi}}\left(\hat{\mathbf{a}}\mid\hat{s}_0,\hat{\boldsymbol{e}}(\boldsymbol{\eta})\right)\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0,\hat{\mathbf{a}},\eta_i\right) \\ =&\sum_{\hat{\mathbf{a}}}\left(\left(\nabla_{\eta_i}\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\right)\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0,\hat{\mathbf{a}},\eta_i\right)+\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\nabla_{\eta_i}\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0,\hat{\mathbf{a}},\eta_i\right)\right) \\ =&\sum_{\hat{\mathbf{a}}}\left(\left(\nabla_{\eta_i}\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\right)\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0,\hat{\mathbf{a}},\eta_i\right)+\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\nabla_{\eta_i}\left(\hat{r}_i-\right.\right. \\ &\qquad\left.\left.\alpha\|W_{\eta_i}^i-W_k^i\|_2^2+\gamma\sum_{\hat{s}'}P\left(\hat{s}'\mid\hat{s}_0,\hat{\mathbf{a}}\right)\tilde{V}_i^{\boldsymbol{\pi}}\left(\hat{s}',\eta_i\right)\right)\right) \\ =&\sum_{\hat{\mathbf{a}}}\left(\left(\nabla_{\eta_i}\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\right)\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}\left(\hat{s}_0,\hat{\mathbf{a}},\eta_i\right)-2\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\alpha\nabla_{\eta_i}W_{\eta_i}^i\left(W_{\eta_i}^i-W_k^i\right)+\right. \\ &\qquad\left.\gamma\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\sum_{\hat{s}'}P\left(\hat{s}'\mid\hat{s}_0,\hat{\mathbf{a}}\right)\nabla_{\eta_i}\tilde{V}_i^{\boldsymbol{\pi}}\left(\hat{s}',\eta_i\right)\right) \\ =&\sum_{x}\sum_{k=0}^{T}P\left(\hat{s}_0\to x,k,\hat{\boldsymbol{\pi}}\right)\gamma^k\sum_{\hat{\mathbf{a}}}\left(\nabla_{\eta_i}\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(x,\hat{\mathbf{a}},\eta_i)-2\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\alpha\nabla_{\eta_i}W_{\eta_i}^i\left(W_{\eta_i}^i-W_k^i\right)\right) \\ =&\sum_{\hat{s}}d^{\hat{\boldsymbol{\pi}}}(\hat{s})\sum_{\hat{\mathbf{a}}}\left(\nabla_{\eta_i}\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(\hat{s},\hat{\mathbf{a}},\eta_i)-2\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\alpha\nabla_{\eta_i}W_{\eta_i}^i\left(W_{\eta_i}^i-W_k^i\right)\right) \\ =&\sum_{\hat{s}}d^{\hat{\boldsymbol{\pi}}}(\hat{s})\sum_{\hat{\mathbf{a}}}\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\left(\nabla_{\eta_i}\log\hat{\boldsymbol{\pi}}_{\hat{\boldsymbol{\theta}}}\tilde{Q}_i^{\hat{\boldsymbol{\pi}}}(\hat{s},\hat{\mathbf{a}},\eta_i)-2\alpha\nabla_{\eta_i}W_{\eta_i}^i\left(W_{\eta_i}^i-W_k^i\right)\right) \\ =&\mathbb{E}_{\hat{\boldsymbol{\pi}}}\left[\left(\sum_{j=1}^{N}\nabla_{\boldsymbol{\eta}}\log\hat{\pi}_{\hat{\theta}_j}^j\left(\hat{a}_j\mid\hat{o}_j,\hat{e}_j(\boldsymbol{\eta})\right)\right)\tilde{Q}^{i,\hat{\boldsymbol{\pi}}}(\hat{s},\hat{\mathbf{a}},\eta_i)-2\alpha\nabla_{\eta_i}W_{\eta_i}^i\left(W_{\eta_i}^i-W_k^i\right)\right]. \end{aligned} \tag{24}$$

Hence descending a stochastic estimate of this gradient is equivalent to minimizing the following loss:

$$-\sum_{t=0}^{T}\sum_{j=1}^{N}\log\pi_{\hat{\theta}^j}^j\left(\hat{a}_j^t\mid\hat{o}_j^t,\hat{e}_j^t(\boldsymbol{\eta})\right)\sum_{\ell=t}^{T}\gamma^{\ell-t}\left(\hat{r}_i^\ell-\alpha\|\Delta^{i,\ell}(W,k)\|_2^2\right)-2\alpha\nabla_{\eta_i}W_{\eta_i}^{i,t}\Delta^{i,t}(W,k), \tag{25}$$

where $\Delta^{i,\ell}(W,k):=W_{\eta_i}^{i,\ell}-W_k^{i,\ell}$ and $\Delta^{i,t}(W,k):=W_{\eta_i}^{i,t}-W_k^{i,t}$.

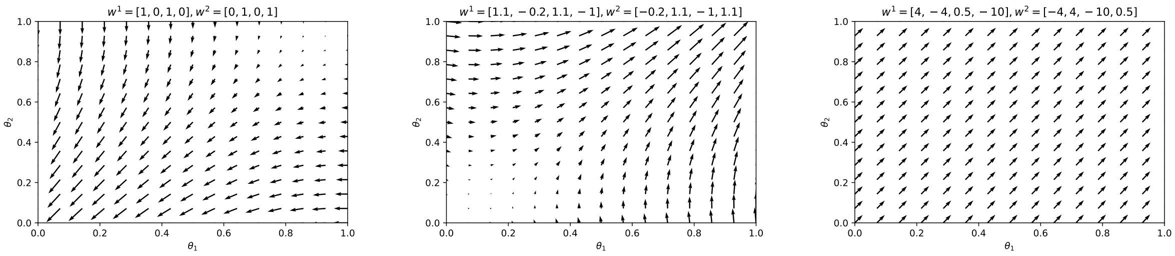
# C Analysis in Iterated Prisoner's Dilemma

Table 2: Prisoner's Dilemma.

| A1/A2 | C | D |
|---|---|---|
| C | $(-1,-1)$ | $(-3,0)$ |
| D | $(0,-3)$ | $(-2,-2)$ |

Similar to Yang et al. (2020), we conduct a complete analysis (Appendix C) using closed-form gradient descent without policy gradient approximation in repeated matrix games. In the stateless Iterated Prisoner's Dilemma (IPD), for example, with payoff matrix in Table 2. Figure 18 shows the vector field of the two agents' cooperation probabilities after setting the probability of the two agents taking a cooperative action at 0.5 and updating the policy using RESVO. As can be seen from the figure, RESVO can converge to mutual cooperation.

**PropositionifnotemptyProposition upn1** *RESVO converges to mutual cooperation in the Iterated Prisoner's Dilemma.*



(a) 1-th round update.

(b) 2-th round update.

(c) $k$-th round update.

Figure 18: Vector fields of the cooperation probability of each agent at the different rounds.

*Proof.* We prove this by deriving closed-form expressions for the updates to parameters of policies and incentive functions. Let $\theta_i$ for $i \in \{1, 2\}$ denote each agent's probability of taking the cooperative action. Let $w^1 := \left[w^{11}, w^{12}\right]$, $w^{11} := \left[w_C^{11}, w_D^{11}\right] \in \mathbb{R}^2$ and $w^{12} := \left[w_C^{12}, w_D^{12}\right] \in \mathbb{R}^2$ denote agent 1's orientation function, where the ratios are given to itself and agent 2 respectively when agent 2 takes action $a_2 = C$ or $a_2 = D$. Similarly, let $w^1$, $w^{21}$ and $w^{22}$ denote agent 2's orientation function. The value function for each agent is defined by

$$\begin{aligned} V_i\left(\theta_1, \theta_2\right) &= \sum_{t=0}^{T} \gamma^t p^T r_i \approx \frac{1}{1-\gamma} p^T r_i, \\ &\text{where } p = \left[\theta_1\theta_2, \theta_1\left(1-\theta_2\right), \left(1-\theta_1\right)\theta_2, \left(1-\theta_1\right)\left(1-\theta_2\right)\right]. \end{aligned} \tag{26}$$

The total reward received by each agent is

$$\begin{aligned} r_1 &= \left[-w_C^{11} - w_C^{21}, -3w_D^{11} + 0, 0 - 3w_D^{21}, -2w_D^{11} - 2w_D^{21}\right], \\ r_2 &= \left[-w_C^{22} - w_C^{12}, 0 - 3w_D^{12}, -3w_D^{22} + 0, -2w_D^{22} - 2w_D^{12}\right]. \end{aligned} \tag{27}$$

Agent 2 updates its policy via the update

$$\begin{aligned}\hat{\theta}_2 =&\theta_2 + \zeta \nabla_{\theta_2} V_2 \left(\theta^1, \theta^2\right)\\ =&\theta_2 + \frac{\zeta}{1-\gamma} \nabla_{\theta_2} \left(\theta_1 \theta_2 \left(-w_C^{22} - w_C^{12}\right) - \theta_1 \left(1-\theta_2\right) 3 w_D^{12}\right.\\ &\left. - \left(1-\theta_1\right) \theta_2 3 w_D^{22} + \left(1-\theta_1\right)\left(1-\theta_2\right)\left(-2w_D^{22} - 2w_D^{12}\right)\right)\\ =&\theta_2 + \frac{\zeta}{1-\gamma} \nabla_{\theta_2} \left[\left(w_D^{12} + w_D^{22} - w_C^{12} - w_C^{22}\right)\theta_1\theta_2 + \left(2w_D^{22} - w_D^{12}\right)\theta_1 + \right.\\ &\qquad \left.\left(2w_D^{12} - w_D^{22}\right)\theta_2 - 2\left(w_D^{12} + w_D^{22}\right)\right]\\ =&\theta_2 + \frac{\zeta}{1-\gamma}\left[\left(3w_D^{22} - w_C^{12} - w_C^{22}\right)\theta_1 - 2\left(w_D^{12} + w_D^{22}\right)\right].\end{aligned} \tag{28}$$

And likewise, for agent 1:

$$\begin{aligned}\hat{\theta}_1 =&\theta_1 + \zeta \nabla_{\theta_1} V_1 \left(\theta^1, \theta^2\right)\\ =&\theta_1 + \frac{\zeta}{1-\gamma} \nabla_{\theta_1} \left(\theta_1 \theta_2 \left(-w_C^{11} - w_C^{21}\right) - \theta_1 \left(1-\theta_2\right) 3 w_D^{11}\right.\\ &\left. - \left(1-\theta_1\right) \theta_2 3 w_D^{21} + \left(1-\theta_1\right)\left(1-\theta_2\right)\left(-2w_D^{11} - 2w_D^{21}\right)\right)\\ =&\theta_1 + \frac{\zeta}{1-\gamma} \nabla_{\theta_1} \left[\left(w_D^{11} + w_D^{21} - w_C^{11} - w_C^{21}\right)\theta_1\theta_2 + \left(2w_D^{21} - w_D^{11}\right)\theta_1 + \right.\\ &\qquad \left.\left(2w_D^{11} - w_D^{21}\right)\theta_2 - 2\left(w_D^{11} + w_D^{21}\right)\right]\\ =&\theta_1 + \frac{\zeta}{1-\gamma}\left[\left(3w_D^{11} - w_C^{11} - w_C^{21}\right)\theta_2 - 2\left(w_D^{11} + w_D^{21}\right)\right].\end{aligned} \tag{29}$$

Let $\hat{p}$ denote the joint action probability under updated policies $\hat{\theta}_1$ and $\hat{\theta}_2$, let $\Delta_2 := \left[\left(3w_D^{22} - w_C^{12} - w_C^{22}\right)\theta_1 - 2\left(w_D^{12} + w_D^{22}\right)\right] \zeta$ $\gamma)$ denote agent 2's policy update and let $\Delta_1 := \left[\left(3w_D^{11} - w_C^{11} - w_C^{21}\right)\theta_2 - 2\left(w_D^{11} + w_D^{21}\right)\right] \zeta/(1-\gamma)$ denote agent 1's policy update. Agent 1 updates its orientation function parameters via

$$\begin{aligned} &w^1 \leftarrow w^1 + \beta \nabla_{w^1} \frac{1}{1-\gamma} \hat{p}^T r_1^{\text{env}}\\ &=w^1 + \frac{\beta}{1-\gamma} \nabla_{w^1} \left[-\left(\theta_1 + \Delta_1\right)\left(\theta_2 + \Delta_2\right) - 3\left(\theta_1 + \Delta_1\right)\left(1 - \theta_2 - \Delta_2\right)\right.\\ &\quad \left. -2\left(1 - \theta_1 - \Delta_1\right)\left(1 - \theta_2 - \Delta_2\right)\right]\\ &=w^1 + \frac{\beta}{1-\gamma} \nabla_{w^1} \left[5\theta_1 - \Delta_1 - 2 + 2\theta_2 + 2\Delta_2\right]\\ &=w^1 + \frac{\beta}{1-\gamma} \begin{bmatrix} \nabla_{w_C^{11}}\left(-\Delta_1\right)\\ \nabla_{w_C^{12}}\left(2\Delta_2\right)\\ \nabla_{w_D^{11}}\left(-\Delta_1\right)\\ \nabla_{w_D^{12}}\left(2\Delta_2\right)\end{bmatrix} = w^1 + \frac{\beta}{1-\gamma} \begin{bmatrix} \frac{\zeta}{1-\gamma}\theta_2\\ -\frac{2\zeta}{1-\gamma}\theta_1\\ -\frac{\zeta}{1-\gamma}\left(3\theta_2 - 2\right)\\ -\frac{4\zeta}{1-\gamma}\end{bmatrix}\\ &=w^1 + \frac{\zeta\beta}{(1-\gamma)^2}\begin{bmatrix}\theta_2\\ -2\theta_1\\ -(3\theta_2 - 2)\\ -4\end{bmatrix}.\end{aligned} \tag{30}$$

By symmetry, agent 2 updates its orientation function parameters via

$$\begin{aligned} w^2 \leftarrow\; & w^2 + \beta \nabla_{w^2} \frac{1}{1-\gamma} \hat{p}^T r_2^{\text{env}} \\ =& w^2 + \frac{\beta}{1-\gamma} \nabla_{w^2} \left[ -\left(\theta_1 + \Delta_1\right)\left(\theta_2 + \Delta_2\right) - 3\left(1 - \theta_1 - \Delta_1\right)\left(\theta_2 + \Delta_2\right) \right. \\ & \left. -2\left(1 - \theta_1 - \Delta_1\right)\left(1 - \theta_2 - \Delta_2\right) \right] \\ =& w^2 + \frac{\beta}{1-\gamma} \nabla_{w^2} \left[ -\theta_2 - \Delta_2 - 2 + 2\theta_1 + 2\Delta_1 \right] \\ =& w^2 + \frac{\beta}{1-\gamma} \begin{bmatrix} \nabla_{w_C^{21}} \left(2\Delta_1\right) \\ \nabla_{w_C^{22}} \left(-\Delta_2\right) \\ \nabla_{w_D^{21}} \left(2\Delta_1\right) \\ \nabla_{w_D^{22}} \left(-\Delta_2\right) \end{bmatrix} = w^2 + \frac{\beta}{1-\gamma} \begin{bmatrix} -\frac{2\zeta}{1-\gamma}\theta_2 \\ \frac{\zeta}{1-\gamma}\theta_1 \\ -\frac{4\zeta}{1-\gamma} \\ -\frac{\zeta}{1-\gamma}\left(3\theta_1 - 2\right) \end{bmatrix} \\ =& w^2 + \frac{\zeta\beta}{(1-\gamma)^2} \begin{bmatrix} -2\theta_2 \\ \theta_1 \\ -4 \\ -(3\theta_1 - 2) \end{bmatrix}. \end{aligned} \tag{31}$$

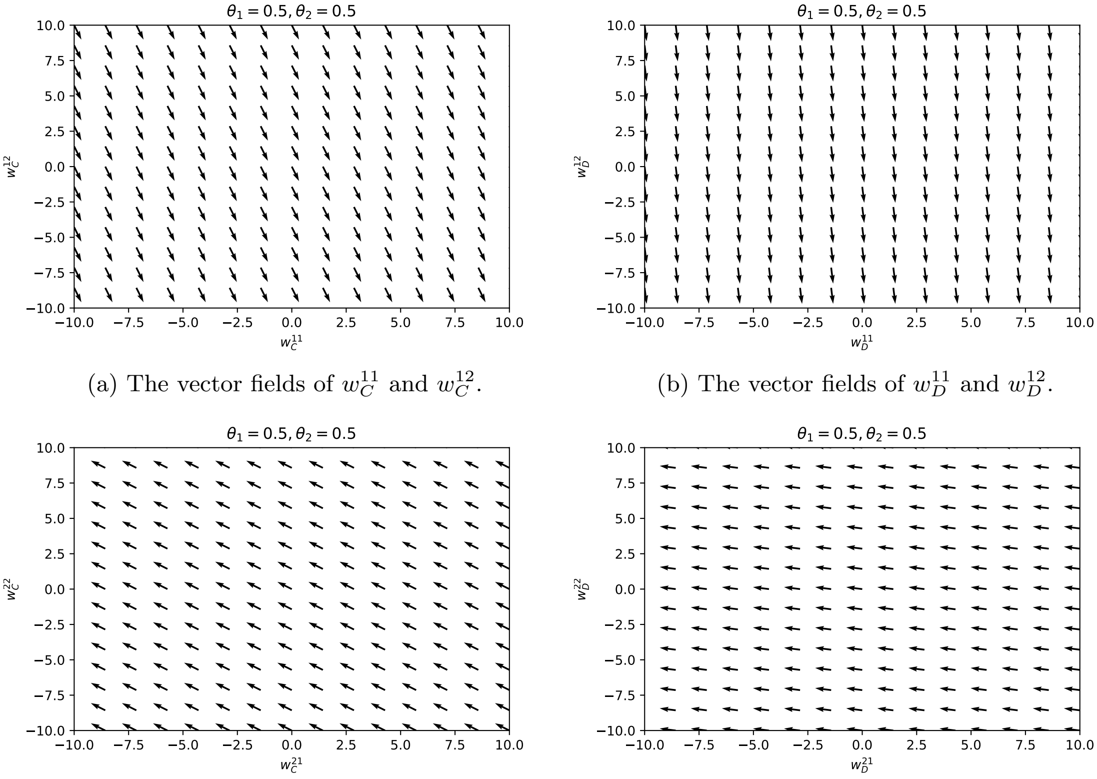
It can be seen from (30) and (31) that in the Prisoner's Dilemma shown in Table 2, except for $w_C^{11}$ and $w_C^{22}$, which are constantly increasing, other terms are constantly decreasing. This ensures that $\theta_1$ and $\theta_2$ in (28) and (29) will eventually continue to increase, that is, the two agents converge to mutual cooperation. Of course, in the above derivation, we ignore the low-rank and mutual information constraints on $w^1$ and $w^2$. We believe that as long as the parameters are adjusted reasonably so that the direction of $w^1, w^2$ update is consistent with the above derivation process, the algorithm can be guaranteed to converge to mutual cooperation.

□

Below we visualize the updating direction of each parameter in the form of the vector field. We initialize $\theta_1 = 0.5, \theta_2 = 0.5, w^1 = [1, 0, 1, 0], w^2 = [0, 1, 0, 1]$. After fixing the values of $\theta_1, \theta_2$, the change of the orientation parameters of each agent is shown in Figure 19.

From Figure 19., we can see that different parameters update at different speeds. According to the speed of parameters change in Figure 19., we update $w^1 : [1, 0, 1, 0] \rightarrow [1.1, -0.2, 1.1, -1]$ and $w^2 : [0, 1, 0, 1] \rightarrow [-0.2, 1.1, -1, 1.1]$. Correspondingly, we show the vector fields of $\theta_1$ and $\theta_2$ before and after the orientation parameters update are shown in Figure 20.

It can be seen from Figure 20 that after a round of parameter updates, the policies of the two agents change from mutual defection to mutual cooperation. As the agent orientation parameters and the cooperation probabilities continue to change (Figure 21: $\theta_1$ $[0.5 \rightarrow 0.6], \theta_2$ $[0.5 \rightarrow 0.6]$; Figure 22: $w^1$ $[1.1, -0.2, 1.1, -1] \rightarrow [1.5, -1, 1.2, -3], w^2$ $[-0.2, 1.1, -1, 1.1] \rightarrow [-1, 1.5, -3, 1.2]$; Figure 23: $\theta_1$ $[0.6 \rightarrow 0.8], \theta_2$ $[0.6 \rightarrow 0.8]$; Figure 24: $w^1$ $[1.5, -1, 1.2, -3] \rightarrow [4, -4, 0.5, -10], w^2$ $[-1, 1.5, -3, 1.2] \rightarrow [-4, 4, -10, 0.5]$), we can see that the agents eventually converge to stable mutual cooperation.



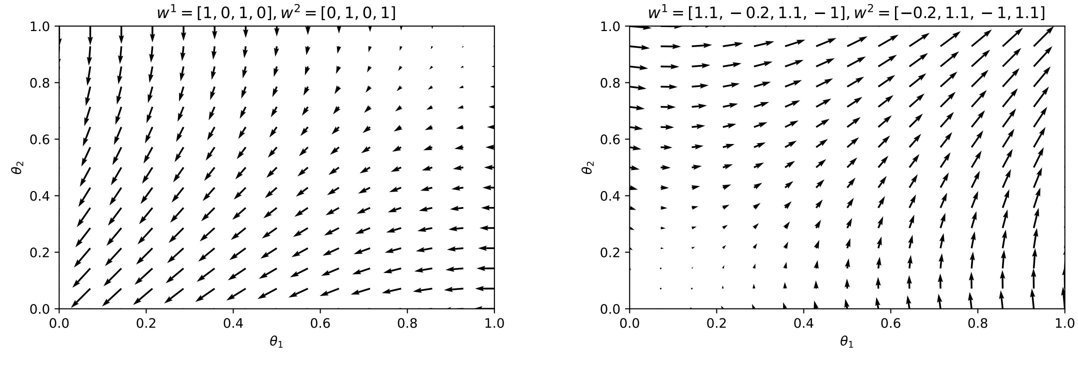
(a) The vector fields of $w_C^{11}$ and $w_C^{12}$.

(b) The vector fields of $w_D^{11}$ and $w_D^{12}$.

(c) The vector fields of $w_C^{21}$ and $w_C^{22}$.

(d) The vector fields of $w_D^{21}$ and $w_D^{22}$.

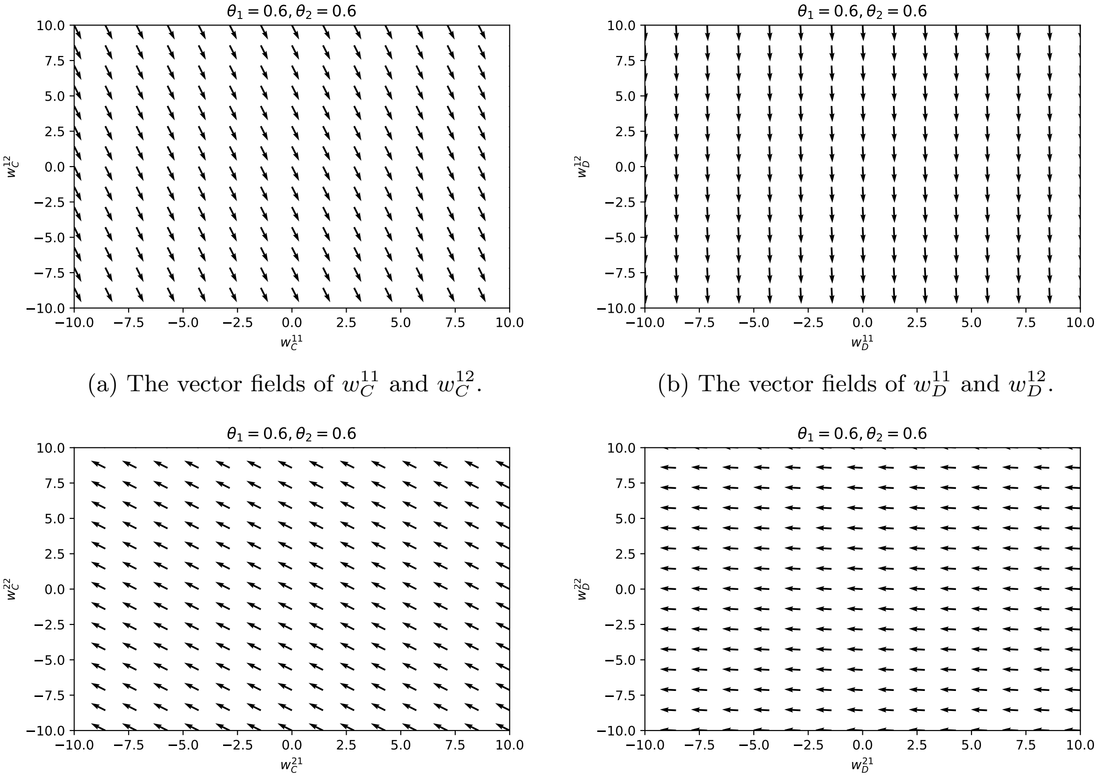
Figure 19: Vector fields of orientation parameters of each agent with $\theta_1 = 0.5, \theta_2 = 0.5$.



(a) Before the orientation parameters update.

(b) After the orientation parameters update.

Figure 20: Vector fields of the probability of cooperation of each agent before and after the orientation parameters update $w^1 : [1, 0, 1, 0] \rightarrow [1.1, -0.2, 1.1, -1], w^2 : [0, 1, 0, 1] \rightarrow [-0.2, 1.1, -1, 1.1]$.



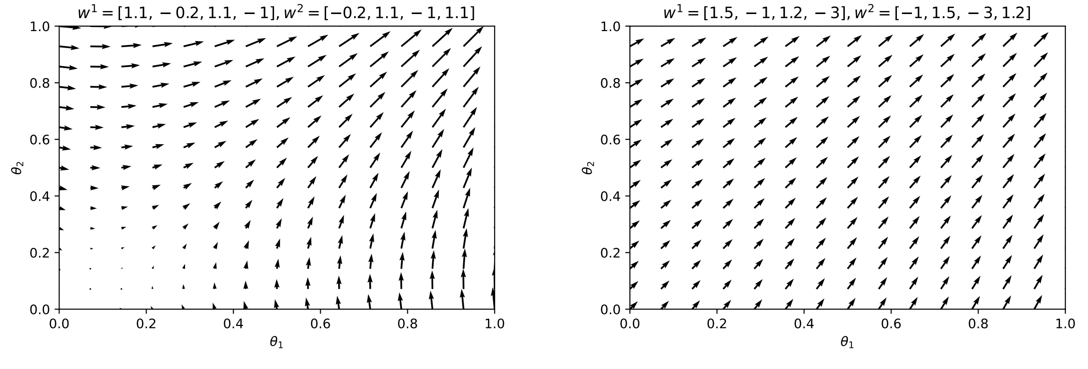
(a) The vector fields of $w_C^{11}$ and $w_C^{12}$.

(b) The vector fields of $w_D^{11}$ and $w_D^{12}$.

(c) The vector fields of $w_C^{21}$ and $w_C^{22}$.

(d) The vector fields of $w_D^{21}$ and $w_D^{22}$.

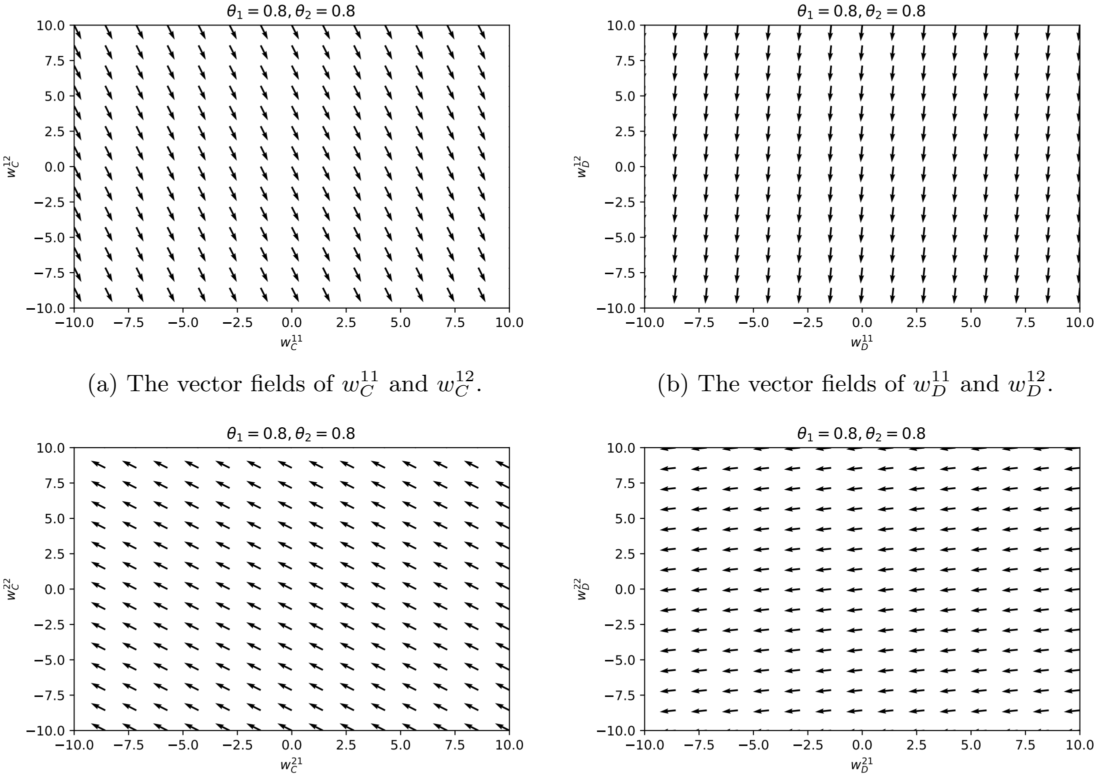
Figure 21: Vector fields of orientation parameters of each agent with $\theta_1 = 0.6, \theta_2 = 0.6$.



(a) Before the orientation parameters update.

(b) After the orientation parameters update.

Figure 22: Vector fields of the probability of cooperation of each agent before and after the orientation parameters update $w^1 : [1.1, -0.2, 1.1, -1] \to [1.5, -1, 1.2, -3], w^2 : [-0.2, 1.1, -1, 1.1] \to [-1, 1.5, -3, 1.2]$.



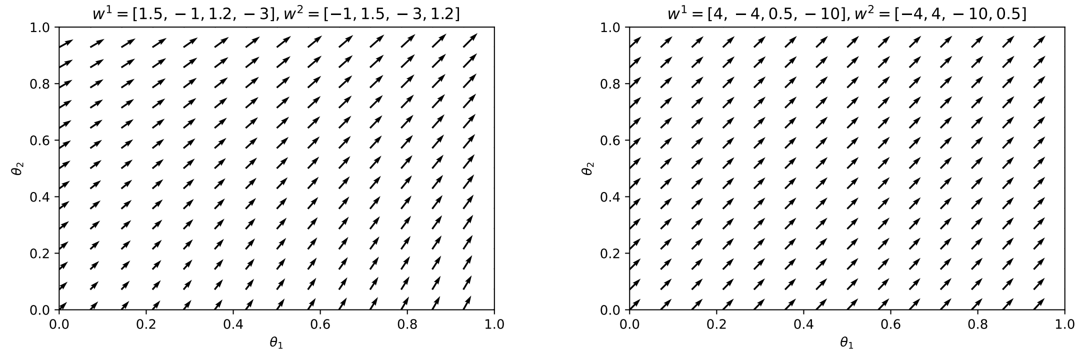
(a) The vector fields of $w_C^{11}$ and $w_C^{12}$.

(b) The vector fields of $w_D^{11}$ and $w_D^{12}$.

(c) The vector fields of $w_C^{21}$ and $w_C^{22}$.

(d) The vector fields of $w_D^{21}$ and $w_D^{22}$.

Figure 23: Vector fields of orientation parameters of each agent with $\theta_1 = 0.8, \theta_2 = 0.8$.



(a) Before the orientation parameters update.

(b) After the orientation parameters update.

Figure 24: Vector fields of the probability of cooperation of each agent before and after the orientation parameters update $w^1 : [1.5, -1, 1.2, -3] \rightarrow [4, -4, 0.5, -10], w^2 : [-1, 1.5, -3, 1.2] \rightarrow [-4, 4, -10, 0.5]$.