applied sciences

MDPI

Article

Adherence Improves Cooperation in Sequential Social Dilemmas

Yuyu Yuan [1,2,*], Ting Guo [1,2], Pengqian Zhao [1,2] and Hongpu Jiang [1]

[1] School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing 100876, China
* Correspondence: yuanyuyu@bupt.edu.cn

Abstract: Social dilemmas have guided research on mutual cooperation for decades, especially the two-person social dilemma. Most famously, Tit-for-Tat performs very well in tournaments of the Prisoner's Dilemma. Nevertheless, they treat the options to cooperate or defect only as an atomic action, which cannot satisfy the complexity of the real world. In recent research, these options to cooperate or defect were temporally extended. Here, we propose a novel adherence-based multi-agent reinforcement learning algorithm for achieving cooperation and coordination by rewarding agents who adhere to other agents. The evaluation of adherence is based on counterfactual reasoning. During training, each agent observes the changes in the actions of other agents by replacing its current action, thereby calculating the degree of adherence of other agents to its behavior. Using adherence as an intrinsic reward enables agents to consider the collective, thus promoting cooperation. In addition, the adherence rewards of all agents are calculated in a decentralized way. We experiment in sequential social dilemma environments, and the results demonstrate the potential for the algorithm to enhance cooperation and coordination and significantly increase the scores of the deep RL agents.

Keywords: multi-agent reinforcement learning; multi-agent system; intrinsic reward; counterfactual reasoning; sequential social dilemmas

check for updates

Citation: Yuan, Y.; Guo, T.; Zhao, P.; Jiang, H. Adherence Improves Cooperation in Sequential Social Dilemmas. *Appl. Sci.* **2022**, *12*, 8004. https://doi.org/10.3390/app12168004

Academic Editor: Jee Hang Lee

Received: 2 July 2022
Accepted: 3 August 2022
Published: 10 August 2022

1. Introduction

Reinforcement learning (RL) has become an increasingly attractive option for adaptive agents to accomplish tasks in complex problem spaces due to its theoretical generalizability [1]. Reinforcement learning shows good performance in many scenarios, such as autonomous driving [2], video games [3], robot control [4], etc. However, single-agent reinforcement learning is not adequate in complex real-world scenarios involving multiple agents. In a multi-agent system, agents not only consider environmental factors, but also need to interact with other agents. Therefore, more researchers have begun to pay attention to the application of reinforcement learning in multi-agent systems.

Multi-agent systems are divided into two categories by whether they are centralized or not to handle the tasks of the agents. First, in a multi-agent system, all agents are assigned tasks or resources in a centralized way [5]. Multi-agent reinforcement learning (MARL) considers all agents as a whole to interact with the environment and directly obtains the optimal joint action [6]. However, with the growth of the number of agents, reinforcement learning will face the problem of the dimensional explosion of state space and action space and difficult to converge. Another type of multi-agent system is where each agent can handle tasks independently [7,8]. However, simply applying the single-agent algorithm makes each agent ignore the actions of the other agents and learn greedy strategies, which leads to the separation between the individual and the collective, and cannot establish an effective cooperative relationship. Especially with the development of society, the Internet and the Internet of Things create new challenges and threats to cooperation within multi-agent systems [9,10]. To sum up, joint learning and independent learning have their advantages and disadvantages. However, more scenarios [11] require agents to be responsible for different tasks to achieve collaboration, which requires them to learn and make decisions independently.

The most challenging scenario is the social dilemma [12,13] in which the Nash equilibrium is not the optimal solution [14]. So far, there has been much research devoted to motivating rational agents to cooperate in social dilemma games, especially repeated matrix Prisoner's Dilemma (PD) games [15,16]. In repeated PD, mutual cooperation is, of course, the ideal long-term strategy, but the short-term benefits of defection are considerable. However, classic repeated PD games fail to capture several key aspects of real-world social dilemma scenarios [17,18]. It is proposed that realistic moves are not simple atomic actions and cannot be easily labeled as cooperation or defection or learned from the payoffs [17]. Instead, cooperation/defection is a sequence of actions or a policy on a graded scale, and the payoffs are usually delayed (e.g., [19,20]). In order to better simulate realistic social interactions, a spatially and temporally extended version of social dilemmas is introduced, that is, sequential social dilemmas (SSDs) [18], which is the type of problem we study in this work.

Intrinsic motivation for RL is the solution to this kind of situation with delayed reward [21,22]. The intrinsic motivation method refers to the allocation of additional intrinsic rewards to individuals, which allows individuals to learn valuable behaviors in various tasks and environments, especially with the lack of environmental rewards [23]. Previous methods of intrinsic motivation usually focus on empowerment obtained by searching for the maximal mutual information (e.g., [24,25]), or curiosity, which can serve as an intrinsic reward to enable the agent to explore its environments and learn skills (e.g., [26,27]). Prosocial learning agents who care about the rewards of their partners can increase the probability that groups converge to good outcomes [28]. However, this method relies on handcrafted rewards specific to the environment. Inequity aversion promotes cooperation in several SSDs environments, and the results help explain how large-scale cooperation emerges and persists [29,30]. However, this method also has the unrealistic problem of allowing agents to access other agents' rewards. Furthermore, Jaques et al. use counterfactual reasoning to calculate social influence as an intrinsic motivation to maximize the collective reward, but this approach leaves some agents constantly exploited [31]. Yuan et al. propose the counterfactual-based action evaluation algorithm, which promotes cooperation by calculating the contribution of actions [32]. However, it employs centralized training to ensure that the agents learn to coordinate.

In conclusion, there is a need for a more practical approach to the SSDs that enables agents to learn to cooperate and guarantees that each agent is independent and not exploited. In the end, we propose a novel adherence-based multi-agent reinforcement learning algorithm for achieving cooperation and coordination by rewarding agents who adhere to other agents. The idea is inspired by social influence, in which each agent calculates its own influence on other agents, and uses the influence value as its extra reward to train itself [31]. When an agent changes its behavior, other agents whose behavior is greatly affected are considered to adhere to the former. When an agent adheres to other agents, it is called a adherent and deserves a reward. During training, each agent calculates the degree of adherence of other agents to its behavior and sends the adherence value to other agents as their reward. The evaluation of adherence is based on counterfactual reasoning [33,34]. The method allows agents to learn independently in multi-agent environments without the need for centralized training. Thus, the proposed algorithm provides us with a simple and effective method to overcome long-term unrealistic assumptions in this research area (i.e., centralized training, sharing parameters, and knowing each other's rewards).

To study our method, we performed some experiments in SSDs environments. Each agent is equipped with an extra internal deep neural network (DNN), which is used to guide behavior and predict the subsequent actions of other agents. The model can learn policies directly from pixels. In addition, the agent can simulate its counterfactual behaviors to evaluate the adherence of other agents to itself. We show that the algorithm can enhance cooperation and coordination in SSDs to break the dilemma and significantly improve the score of deep RL agents.

The rest of this paper is organized as follows: In Section 2, we describe the preliminaries and review the background on partial observation Markov decision process, multi-agent reinforcement learning, and sequential social dilemmas. Section 3 describes the algorithm in detail. Section 4 provides the experimental environments and results, and we discuss the results in Section 5. Finally, Section 6 concludes this paper and gives some suggestions for future research directions.

2. Preliminaries and Backgrounds

Our work concerns sequential social dilemmas in MARL. Therefore, we review the related backgrounds of sequential social dilemmas and MARL methods in this section.

2.1. Partially Observable Markov Games

We consider partially observable Markov games as a framework for multi-agent reinforcement learning [35,36]. At each time step, the agents perform operations to interact with environments and one another. Then, the environments provide feedback in the form of partial observations of the state space and individual rewards. Observations and rewards are used to determine the agents' subsequent actions because agents must learn to maximize their cumulative rewards through experience. We formalize this as follows.

An N-player partially observable Markov game $\mathcal{M}$ is defined on a set of states $\mathcal{S}$. The observation function $\mathcal{O} : \mathcal{S} \times \{1, \dots, N\} \rightarrow \mathbb{R}^d$ specifies the d-dimensional view of each player on the state space. Write $\mathcal{O}^i = \{o^i | s \in \mathcal{S}, o^i = \mathcal{O}(s, i)\}$ to be the observation space of player i. In each state, players take actions a^i from action sets $\mathcal{A}^1, \dots, \mathcal{A}^N$ with $\mathcal{A}^i$ being the action set of player i. The joint action $a^1, \dots, a^N \in \mathcal{A}^1, \dots, \mathcal{A}^N$ changes the state following the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \rightarrow \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ represents the set of discrete probability distributions over $\mathcal{S}$. Each player obtains an individual reward defined by a reward function $r^i : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \rightarrow \mathbb{R}$ for player i.

Through their own experiences of the environment, agents learn an appropriate behavior policy $\pi^i : \mathcal{O}^i \rightarrow \Delta(\mathcal{A}^i)$ (written $\pi(a^i|o^i)$) based on their own observations $o^i = \mathcal{O}(s, i)$ and environmental rewards $r^i(s, a^1, \dots, a^N)$. For simplicity, we define $\vec{a} = (a^1, \dots, a^N)$, $\vec{o} = (o^1, \dots, o^N)$ and $\vec{\pi}(\cdot|\vec{o}) = (\pi^1(\cdot|o^1), \dots, \pi^N(\cdot|o^N))$. For temporal discount factor $\gamma \in [0, 1]$, the goal of each agent is to maximize its long-term payoff $V^i_{\vec{\pi}}(s_o)$ defined as follows:

$$V^i_{\vec{\pi}}(s_o) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(s_t, \vec{a}_t) | \vec{a}_t \sim \vec{\pi}, s_{t+1} \sim \mathcal{T}(s_t, \vec{a}_t)\right] \tag{1}$$

2.2. Sequential Social Dilemmas

Social dilemmas provide a powerful platform to study the development of behavior and the emergence of cooperation. Multi-person social dilemmas are divided into two categories: public goods dilemmas and commons dilemmas [37]. In public goods dilemmas [38,39], every individual has to pay the price in exchange for the resources shared by all. That means that public goods are non-exclusive, so there is a temptation to enjoy products without contribution. Although free-riding is rational for individuals, there will be no public good if everyone does so, and all will be worse off. The commons dilemmas [40–42] is a situation in which an individual's short-term selfish interests are at odds with the long-term group interests. In commons dilemmas, every individual who has open access to resources is tempted to deplete resources shared by all, according to their self-interest. The regeneration rate of resources depends on the number of resources. When the number of resources is reduced to a certain level, the resources are no longer regenerated.

In social dilemmas, there are four possible outcomes of each player, namely R (reward arising from mutual cooperation), P (punishment of mutual defection), S (sucker outcome obtained by the player who cooperates with a defecting one), and T (temptation outcome achieved by defecting against a cooperator). The four payoffs must satisfy the following *social dilemma inequalities* [13]:

- $R > P$: mutual cooperation is preferred to mutual defection.
- $R > S$: mutual cooperation is preferred to being exploited by a defector.
- $2R > S + T$: mutual cooperation is preferred to an equal probability of unilateral cooperation and defection.
- $T > R$: exploiting a cooperator is preferred over mutual cooperation.
- $P > S$: mutual defection is preferred over being exploited.

In order to capture aspects of real-world social dilemmas, Leibo et al. proposed sequential social dilemmas (SSDs) [18] which are spatially and temporally extended partially observable Markov games, because practical cooperation and defection are not simple atomic actions. Markov games are environments that simulate the Markov decision process and are convenient for studying reinforcement learning. An N-player SSD is a tuple $(\mathcal{M}, \Pi_c, \Pi_d)$ of Markov games and satisfies the following properties. Π_c and Π_d are two disjoint sets of policies which represent cooperation and defection, respectively. The set of all players' policies is $(\pi_c^1, \ldots, \pi_c^l, \pi_d^1, \ldots, \pi_d^m) \in \Pi_c^l \times \Pi_d^m$ with $l + m = N$. Assuming $N = 2$, the empirical payoffs (R, P, S, T) under the initial state s can be defined as $(R(s), P(s), S(s), T(s))$ through their long-term payoff, where

$$R(s) := V_1^{\pi_c,\pi_c}(s) = V_2^{\pi_c,\pi_c}(s), \tag{2}$$

$$P(s) := V_1^{\pi_d,\pi_d}(s) = V_2^{\pi_d,\pi_d}(s), \tag{3}$$

$$S(s) := V_1^{\pi_c,\pi_d}(s) = V_2^{\pi_d,\pi_c}(s), \tag{4}$$

$$T(s) := V_1^{\pi_d,\pi_c}(s) = V_2^{\pi_c,\pi_d}(s). \tag{5}$$

Let the empirical payoffs (R, P, S, T) be induced by policies $(\pi_c^1, \ldots, \pi_c^l, \pi_d^1, \ldots, \pi_d^m) \in \Pi_c^l \times \Pi_d^m$ via Equations (2)–(5). A Markov game is an SSD game if there exists a state $s \in S$ for which the induced payoff satisfies the five *social dilemma inequalities*.

Sequential social dilemmas show the tension between individual short-term incentives and collective long-term interests. An individual agent needs to balance its short-term benefits and long-term benefits. Each agent will get a higher reward in the long-term if all agents choose to cooperate. However, an agent can obtain a higher reward in the short term by choosing to defect with non-cooperative behavior.

Agents are tempted to defect because of the incentive structure of the SSDs. Although all group members in SSDs prefer the rewards of cooperation with each other, the specific incentive structure pushes groups toward reward-suppressing equilibria. Thus, the collective reward achieved by all agents in SSDs becomes a clear measure to evaluate the degree of the agents who learned to cooperate [29].

Two SSDs environments are described in Section 4. Note that the computational requirements for the equilibrium of SSDs are higher due to the higher complexity associated with sequential structures, which necessitates multi-agent deep reinforcement learning methods. In SSDs environments, not only must agents learn to cooperate, but they must also learn to abstain from defects. It is difficult for traditional RL agents to learn how to coordinate or cooperate to solve these tasks effectively. Thus, these SSDs environments have become challenging benchmark tasks for multi-agent cooperation.

2.3. Multi-Agent Reinforcement Learning

MARL in Markov games is the focus of a large number of works, primarily concerned with learning an optimal behaving policy [17]. Multiple RL agents learn policies through trial-and-error interaction with their common environment. We use a distributed asynchronous advantage actor-critic (A3C) approach [43] and a distributed proximal policy optimization (PPO) approach as learning algorithms to train each agent's policy π^i.

In the A3C approach, we use a deep neural network to maintain a policy π^i (actor) and a value function V^{π^i}(critic). A3C uses the policy to choose actions and the value function to estimate states. In addition, the policy is updated by taking steps in the direction of policy gradients, using the value estimate as a baseline to reduce variance. Gradients are generated asynchronously by multi-independent copies of each agent, running in different instances of the environment at the same time. Explicitly, the gradient is computed by

$$\nabla J(\theta) = \nabla_\theta log\pi_\theta(a_t|s_t;\theta)A(s_t,a_t;\theta,\theta_v), \tag{6}$$

where $A(s_t,a_t;\theta,\theta_v)$ is the advantage function, which estimates the advantage of action a_t in state s_t according to k-step backups, $\sum_{i=0}^{k-1}\gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta_v) - V(s_t;\theta_v)$, where r_{t+i} is our subjective reward. Moreover, in Section 3.2, we decompose the reward into an extrinsic environmental reward and an intrinsic reward that depends on how much the agent adheres to the other agent.

In the PPO approach, our deep neural network maintains a policy with no value function, and the PPO algorithm uses the clip function to limit the magnitude of the gradient update while ensuring the effective update of the policy. The objective function corresponding to the gradient is

$$J(\theta_k,\theta) = min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}A^{\pi_{\theta_k}}(s_t,a_t), clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)A^{\pi_{\theta_k}}(s_t,a_t)\right), \tag{7}$$

in which ϵ is a hyperparameter, which roughly controls the distance between the new policy and the old policy.

3. Method

In this section, we fully introduce the algorithm we proposed and its optimization. Section 3.1 outlines the main components of the algorithm. Section 3.2 details the method of the adherence calculation, and Section 3.3 introduces the models we used. Section 3.4 presents the reward method that the algorithm utilizes. Section 3.5 further describes how the algorithm is optimized.

3.1. Overview

Due to the specific incentive structure of SSDs, standard RL agents usually choose to fight separately. To achieve cooperation, multiple agents need to evolve in the direction of cooperation simultaneously. Intuitively, the possibility of all agents' policies' convergence to cooperation is essentially slight in the entire policy space [17]. Therefore, a method is needed to ensure that the agent evolves toward cooperative behavior. To this end, inspired by social influence, we propose a adherence-based multi-agent reinforcement learning algorithm, which encourages agents to adhere to others by rewarding agents.

In the method, if one agent's actions can be influenced by another agent's behaviors, it means that the former adheres to the latter. We define the former as adherent and are rewarded by the latter based on adherence. In order to obtain more rewards, each agent needs to actively adhere to other agents. Simultaneously, each agent has to judge whether other agents are adherent and reward them to strengthen their adherence to itself. In our paper, the method can transform the selfish behaviors of agents into the desired cooperative behaviors. The algorithm mainly includes three essential parts:

- Adherence evaluation: To cooperate better, each agent needs to understand the behavior of other agents better and judge whether it is adherent before deciding to cooperate. For knowing the adherence of other agents, it is necessary to calculate their adherence value concretely. Therefore, in Section 3.2, we illustrate how the adherence is calculated.
- Adherence evaluation model: The premise of adherence is to know each other's behavior, but agents do not have the right to access the actions of other agents, so we set that agents need to predict the behaviors of other agents. In order to know other

agents, we equip each agent with an adherence evaluation model (AEM). The model can be used not only to guide the behavior of the agent owner, but also to predict the next actions of other agents. The detailed introduction of the model is in Section 3.3.

- Reward design: Based on the above two parts, we can design the rewards for agents. In the method, the designed reward will be handed over to the agent for training, which can transform the dilemma state of SSDs into a cooperative state. The calculation of the reward is divided into two steps. The first step is to calculate the adherence value through predictions, and the second step is to reward the corresponding agent based on the adherence.

3.2. Adherence Evaluation

When an agent changes its actions due to the actions of another agent, it means that it adheres to that agent. The agent will obtain extra reward for adhering to another agent's actions. Therefore, we need to know the exact adherence value in order to train agents better. The adherence value represents the degree of adherence that an agent has to another agent.

To compute the adherence of one agent to another, we suppose that there are two agents, k and j, and agent j adjusts its policy based on agent k's action at time t, a_t^k, that is, j observes k's actions at time t and then choose his own actions. Therefore, the probability of agent j's next action is $p(a_t^j|a_t^k, o_t^j)$, which is called the conditional policy. Then, by replacing a_t^k with a counterfactual action, $\tilde{a}_t^k$, we can calculate a new distribution over j's next action, $p(a_t^j|\tilde{a}_t^k, o_t^j)$. Essentially, it is as if agent k asked a question: "what will j do if I choose a different action in this situation?". The difference between the distributions indicates whether j will change its action if k changes its action.

By multiplying the probabilities of k's all counterfactual behaviors by the corresponding distribution of j, and summing them up, we can obtain the marginal policy of j,

$$p(a_t^j|o_t^j) = \sum_{\tilde{a}_t^k} p(a_t^j|\tilde{a}_t^k, s_t^j)p(\tilde{a}_t^k|o_t^j). \tag{8}$$

In other words, we obtain a j policy that does not take agent k into account at all. The difference between the conditional policy of j given k's action and the marginal policy of j is a measure of the adherence of j to k. It represents the degree to which j changes the distribution of planned actions due to the action of k. That is, it shows how much j adheres to k. Here we calculate the difference between policies by KL divergence. Thus, j's adherence to k is evaluated as follows:

$$c_t = D_{KL}[p(a_t^j|a_t^k, o_t^j)\|p(a_t^j|o_t^j)]. \tag{9}$$

According to the Equation (9), the total adherence reward of j is

$$c_t^j = \sum_{k=0,k\neq j}^{N} \Big[D_{KL}[p(a_t^j|a_t^k, o_t^j)\|p(a_t^j|o_t^j)]\Big]. \tag{10}$$

The main procedure of adherence evaluation is shown in Algorithm 1.

3.3. Adherence Evaluation Model

Calculating the adherence reward as introduced in the previous section requires knowing the probability of the actions of other agents; it requires that agent k chooses its action before j. Therefore, j can calculate the adherence to k, but k cannot calculate the adherence to j. Hence, it is less realistic than a scenario in which each agent is trained independently and can adhere to each other.

For agents to be able to train independently, we relax that assumption by introducing an adherence evaluation model (AEM) equipped with a prediction module. The model can avoid the problem of recursive interdependence. At each timestep, each agent chooses an action; these actions are concatenated into a joint action vector $a_t = [a_t^0, a_t^1, \ldots, a_t^N]$, for N agents. This action vector a_t is then given as input to every agent. Based on their actions and the agent's self-view of the state, the prediction module is trained to predict the next actions of all other agents, $p(a_{t+1}|a_t, o_t^i)$.

Algorithm 1 Evaluation of j's adherence value

```
Input: $N$, agent set
       $o_t^j$, agent $j$'s observation of the state at time $t$
       $p$, agent $j$'s policy
       $\mathcal{A}^k$, agent $k$'s action set $T$
       $a_t^k$, agent $k$'s action at time $t$
Output: $c_t^j$, the overall adherence value of all other agents
$c_t^j = 0$
for each agent $k \in N$ do
    if $k$ is not $j$ then
        $p(a_t^j|o_t^j) = 0$
        $c_t = 0$
        for each $\tilde{a}_t^k \in \mathcal{A}^k$ do
            Replace counterfactual action, calculate marginal policy
            $p(a_t^j|o_t^j) = p(a_t^j|o_t^j) + p(a_t^j|\tilde{a}_t^k, o_t^j)p(\tilde{a}_t^k|o_t^j)$
        end for
        Calculate $j$'s adherence to $k$ $c_t = D_{KL}[p(a_t^j|a_t^k, o_t^j) \| p(a_t^j|o_t^j)]$
    end if
    Calculate $j$'s total adherence reward $c_t^j = c_t^j + c_t$
end for

return $c_t^j$
```

As shown in Figure 1, the bottom half of the model is the prediction module and the top half is a standard RL module equipped with a RL head. The RL head is trained to learn a behavior policy π_e, and in the A3C approach, the value function V_e is also learned. To extract the latent state of the environment and enhance the long-term memory of the agent, we connect these two modules to a convolutional layer and equip each of them with a long short-term memory (LSTM) recurrent layer [44]. Moreover, the prediction model is essentially a supervised model, mainly trained by observing other agents' action trajectories.

3.4. Reward Design

The adherence is an invisible driving force that can transform the state of social dilemma into a state of mutual cooperation. Specifically, the immediate reward of an agent is modified to $r_t^k = \alpha e_t^k + \beta c_t^k$, where e_t^k is the environmental reward, and c_t^k is the adherence reward. In the experiment, only environmental rewards are compared.

Agents receive the immediate reward in the following way. First, we use the trained AEM of each agent to compute the adherence value. Each agent k imagines possible counterfactual actions and inputs these actions into its own internal prediction model to predict the next action of other agents at each time step, $p(a_{t+1}|a_t^{-k}, \tilde{a}_t^k, o_t^i)$. By replacing all possible counterfactual actions, the marginal policies for all other agents are derived, $p(a_{t+1}|a_t^{-k}, o_t^i)$. Then each agent k can evaluate the adherence of other agents to itself through KL divergence. Finally, each agent sends a reward to the corresponding agent according to the adherence value. The reward is the intrinsic incentive reward of each agent, which can be used to train its own strategy, but it is not included in the total reward metric.

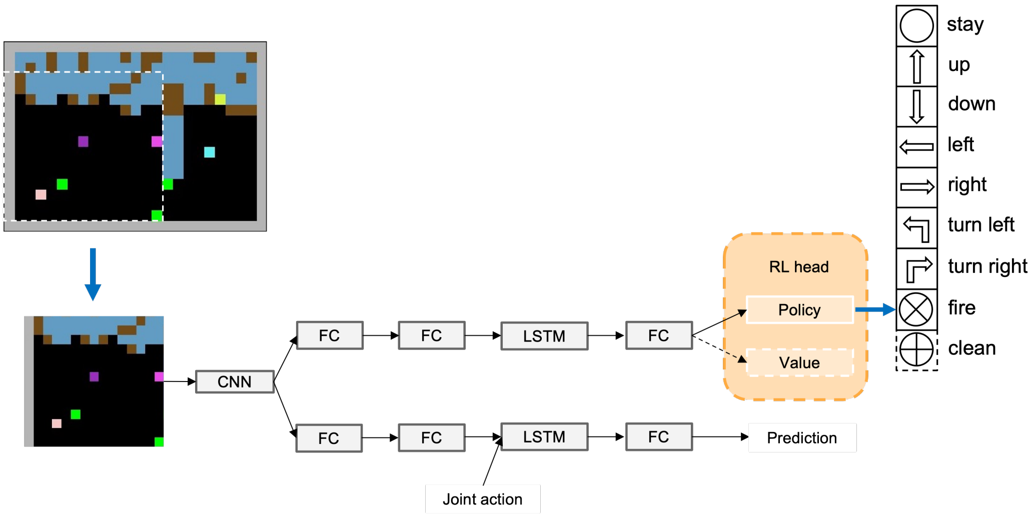

Figure 1. Our network consists of a RL module and a prediction module, which are connected to a convolutional layer. In the baseline, the model contains only the RL module. In the A3C approach, the RL head can learn the RL policy π_e and value function V_e. In the PPO approach, the RL head only learns the policy π_e.

Intuitively, this is similar to how human beings reason about other people's adherence to themselves [45]. Actually, we often ask counterfactual questions, such as "If I did something else in this situation, what would he(she) do?". Then we use our internal prediction models to answer.

3.5. Algorithm Optimization

In order to learn $p(a^j_{t+1}|a^k_t, o^k_t)$, the AEM requires modeling of the behavior of the other agents and the environmental transition function. If the model is inaccurate, it will lead to incorrect estimates of the adherence reward. In particular, when an agent j is not within the field-of-view of agent k, the adherence reward of agent j to agent k is meaningless for the model training of agent k, which is called invalid data. On the contrary, when agent j and agent k can see each other, the adherence reward of agent j to agent k can effectively help the model of agent k to train, which is called valid data.

To this end, we optimize the AEM. We set that agent k only trains its model and gives the adherence reward to agent j when agent j is within agent k's field of view. Only in this way can the evaluation of $p(a^j_{t+1}|a^k_t, o^k_t)$ be more accurate. This restriction may have the side effect of encouraging agents to stay closer. However, considering that humans seek kinship and spend time with others, the internal social rewards that encourage closeness are reasonable [46]. In addition, using this method to obtain more effective data is more helpful to the training of the model.

4. Experiment

In this section, we test the proposed method in two environments (https://github.com/eugenevinitsky/sequential_social_dilemma_games, accessed on 10 March 2022) and describe our results. The two environments are Cleanup and Harvest, which represent different classes of the social dilemma; Cleanup is a public goods dilemma game, while Harvest is a commons dilemma game. These are complex environments with delayed results of actions and partial observability of a complex grid world. The use of these two environments is to test the algorithm of this paper, such that our algorithm can enable agents to learn to coordinate and cooperate in complex environments.

4.1. Environment

Cleanup and Harvest are typical sequential social dilemma games. They are used to verify the effectiveness of the multi-agent reinforcement learning algorithms. Details of the two environments are as follows.

4.1.1. Cleanup

In the cleanup(Figure 2A), a player is rewarded for consuming apples within a 25 × 18 grid-world environment. For every apple consumed by an agent, its reward increases by 1. The growth rate of apples in the orchard is directly proportional to the cleanliness of the nearby river. Over time, the waste in the river increases with a constant probability. When the waste exceeds a certain threshold, the apple growth rate drops to zero. Players can clear the river by firing a cleaning beam. However, the cleaning beam has a limited range, affecting the waste within a short distance in front of players. Thus, players need to maintain the public good of apple regrowth by leaving the orchard to clean the river. In addition, the costly punishment is one of the agents' actions, which is of critical importance in sequential social dilemmas [47,48]. Players can use the penalty beam to punish other players at a small cost (reward-1), and the reward for other players will be reduced by 50. That may be used to discourage free-riding.

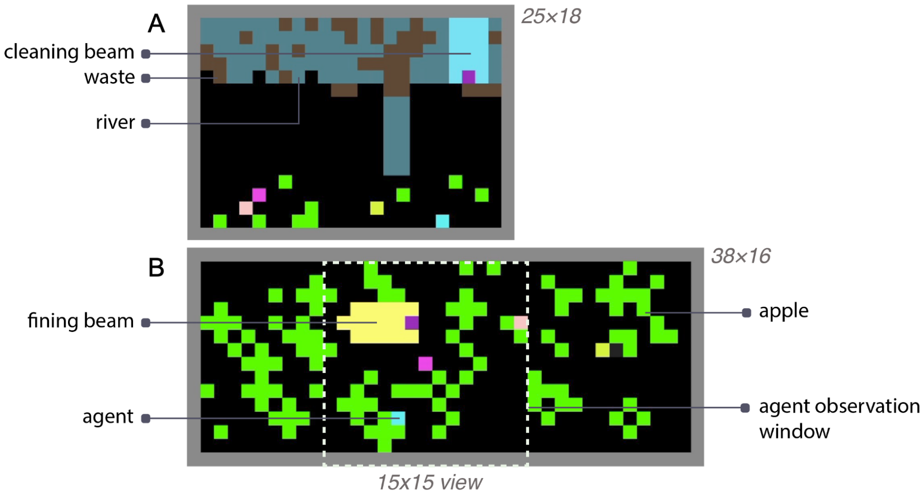

Figure 2. (**A**) Cleanup game. (**B**) Harvest game. Agents obtain rewards by collecting apples which is the green grid. Agents can fine each other through punishment light.

At the beginning of each episode, the environment resets waste just above the saturation point. In order for apples to spawn, agents must first clean up some waste. Each episode has 1000 steps, and the environment is reset to its initial state after the game is over. The observability of each agent is limited to a 15 × 15 RGB window, which is centered on the agent's current position.

In the cleanup environment, the agent will be rewarded by collecting apples. Nevertheless, only by cleaning up the waste can apples be reproduced. However, the agent is not rewarded for cleaning up the waste. The selfish agent chooses only to collect apples. If all agents collect apples, then no agent acquires any reward. That is a public goods dilemma.

4.1.2. Harvest

Our second experimental environment is the harvest game (Figure 2B), where the goal is to consume apples within a 24 × 26 grid world environment. One consumed apple offers one reward to the corresponding agent in the environment. The probability of apples regrowing on different grids is different, depending on the apples' spatial distribution. The more apples there are nearby, the higher the probability. If individual agents employ an exploitative strategy by greedily consuming too many apples, apples may no longer respawn. A group can achieve sustainable harvesting by not consuming ”endangered apples” (the last unharvested apple in their territory). In addition, agents also have the action of punishment in the Harvest. Players can use the penalty beam to punish other players (reward-50) at a small price to themselves (reward-1).

Each episode has 1000 steps, and the environment is reset to its initial state after the game is over. The agent's observability is also limited to a 15 × 15 RGB window centered on the current position of the agent.

The dilemma is that agents who only focus on short-term benefits lead to the depletion of apples. However, the long-term benefits of the group increase if individuals abstain from collecting apples too quickly. Such situations are unstable because the more greedy the agents, the more likely the resources are to be permanently exhausted. In order to obtain a higher total reward, agents must learn to restrain themselves from harvesting apples.

4.2. Social Outcome Metrics

In our paper, we refer to the metrics of [42]. Consider N independent agents. Let $\{r_t^i | t = 1, ..., T\}$ be the sequence of rewards obtained by the i-th agent over an episode of duration T. The return is given by $R^i = \sum_{t=1}^{T} r_t^i$.

The *Utilitarian* metric (U), measures the sum of all rewards obtained by all agents. The *Equality* metric (E) is defined using the Gini coefficient. The *Sustainability* metric (S) is defined as the average time at which the rewards are collected.

$$U = \mathbb{E}\left[\sum_{i=1}^{N} R^i\right], E = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}|R^i - R^j|}{2N\sum_{i=1}^{N} R^i}, S = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} t^i\right] \ where\ t^i = \mathbb{E}[t|r_t^i > 0].$$

4.3. Results and Analysis

In order to verify the effectiveness of the algorithm, we conducted comparative experiments with the baseline in Cleanup and Harvest. The specific results and analysis are as follows.

4.3.1. Cleanup

The results show that the algorithm can indeed increase total reward and encourage agents to cooperate. We also evaluate an additional baseline where agents' models do not have the prediction module in order to isolate the benefits of the algorithm. The results of the simulation for the cleanup can be seen in Figure 3.

At the beginning of training, agents begin training by acting randomly, diffusing through the space, and learning to clean the river. In the experiment, the agent needs to obtain a large number of apples in order to maximize personal benefits. However, in the face of limited apple resources, the agents have to attack each other, which makes the total waste and the equality drop sharply. Then the group falls into a social dilemma. Ultimately, A3C baseline agents are unable to coincidentally avoid the tragedy, and the total reward converges to near 20, step by step. Meanwhile, the PPO baseline agents' total reward is slowly increasing. After applying the adherence-based algorithm, the collective reward of the A3C approach quickly surpasses the baseline and keeps increasing. In the PPO approach, the overall reward is also rising rapidly, rising faster than the baseline. The collective reward of the algorithm (A3C) reached an average of about 140, and the algorithm (PPO) reached about 400.

As shown in Figure 3, with the help of the adherence-based algorithm, in the cleanup game, having a adherence reward ultimately achieves higher collective rewards. The baseline agent fails to attain socially beneficial results in either category of the game. Due to the nature of the SSDs games, we are able to infer that agents that achieve higher collective rewards have learned to cooperate more effectively. We show that the algorithm can resolve sequential social dilemmas by providing a correct reward.

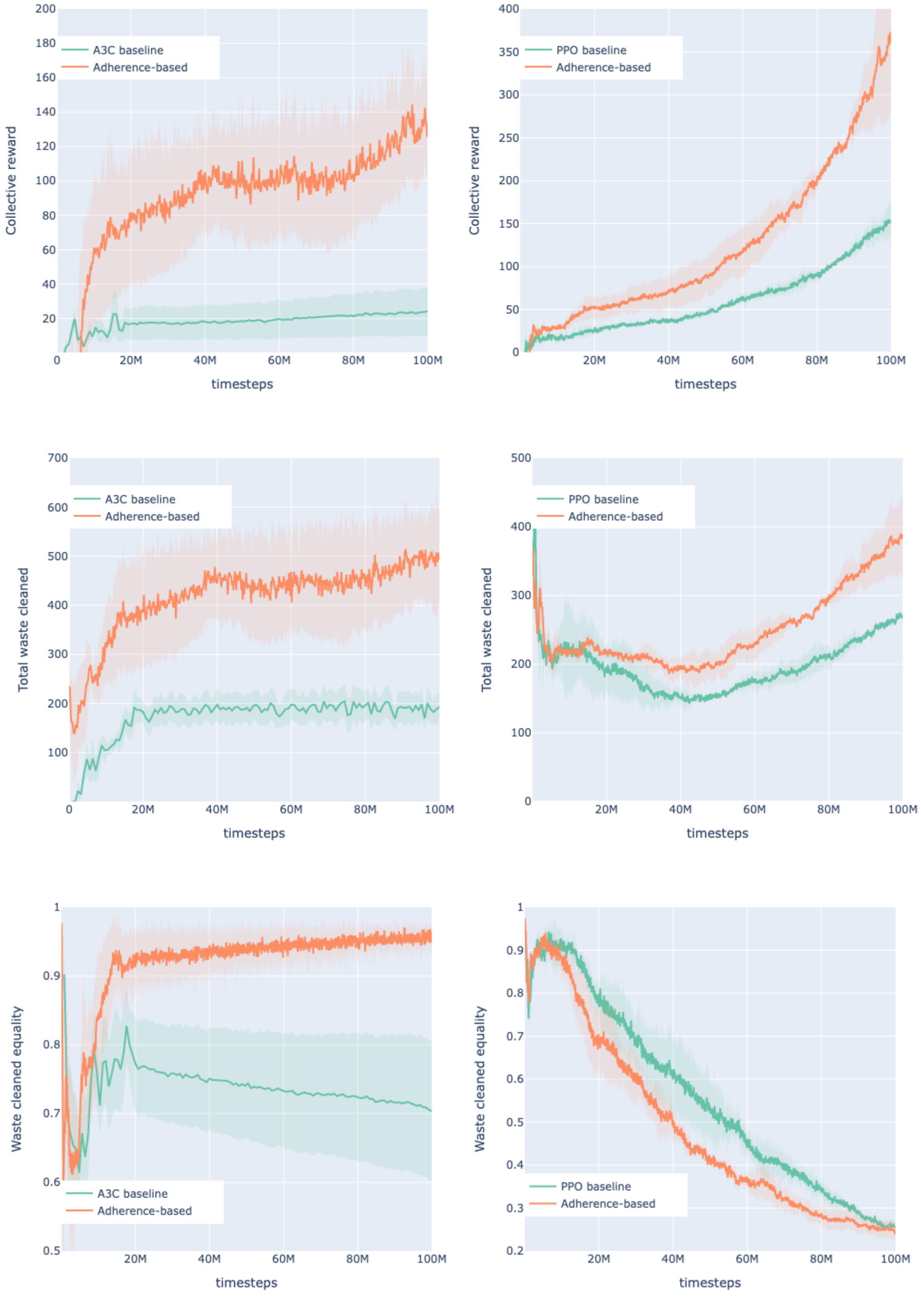

Figure 3. The effect of the adherence-based algorithm on social outcomes in Cleanup. The social outcome metrics are collective reward, total waste cleaned and equality of the waste cleaned, respectively. The left three figures are the results of our method applied to A3C. The right three figures are the experiment results based on PPO.

4.3.2. Harvest

As in the previous subsection, we use the same metrics as the evaluation metric. The baseline method and the adherence-based algorithm are similarly applied to Harvest and produce the results which can be seen in Figure 4.

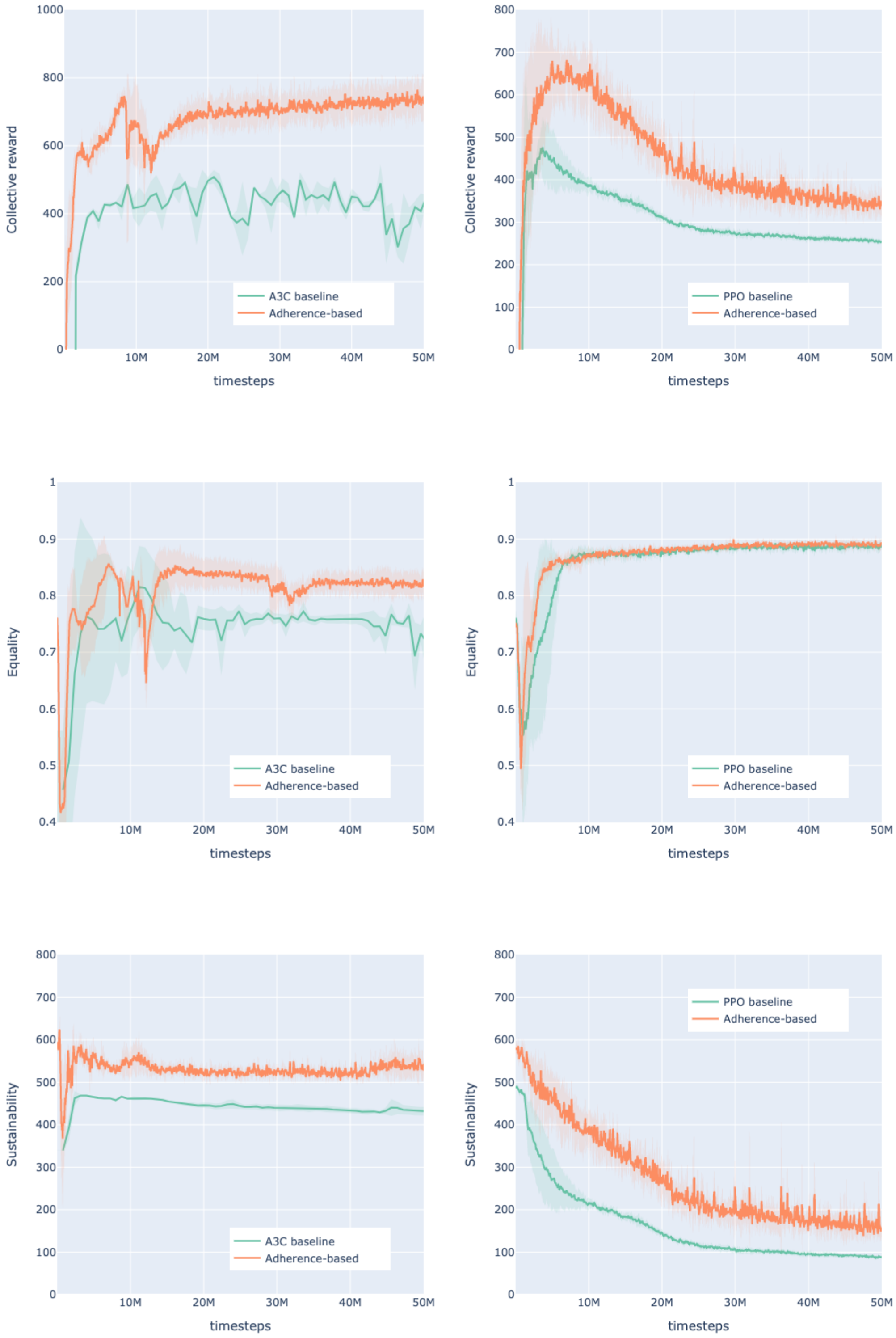

Figure 4. The effect of the adherence-based algorithm on social outcomes in Harvest. The social outcome metrics are collective reward, equality and sustainability, respectively. The left three figures are the results of our method applied to A3C. The right three figures are the experiment results based on PPO.

In the A3C approach, at the beginning of the training, agents explore the environment individually and collect apples whenever they find them. At the same time, they detect that their aggressive behavior does not do any good, so they quickly learn not to use it. As training progresses, the agents learn to move to areas with a greater density of apples in order to harvest rewards more effectively. The collective reward of the baseline can reach 400. The respawn probability of an apple in Harvest is determined by the number of apples around it, so maintaining a reasonable amount of resources allows agents to continue to consume more apples. In the adherence-based algorithm, the collective reward of agents reaches about 800 on the premise that equality is not reduced. Experimental results show that the adherence-based algorithm (A3C) can break sequential social dilemmas by encouraging agents to adhere to each other.

In the PPO approach, the collective rewards of the baseline and the adherence-based algorithm rose rapidly in the short term, but after the 5 M timestep, it began to gradually decline. This happens because agents have learned "too well" how to obtain rewards. With each agent harvesting as quickly as possible, no time is allowed for the apples to recover. Apples quickly become depleted. As a result, the total reward declines precipitously, and the group falls into a social dilemma. In the end, the agents' sustainability was less than half of the random actions at the start of the training. Due to the nature of PPO algorithm, the distance between the new policy and the old is not allowed to be too great. As a result, although the total reward of the adherence-based algorithm is higher than the baseline, it is still unavoidable to fall into social dilemmas.

5. Discussion

In the cleanup game, the agents are too greedy to clean up the river and thus get stuck in a social dilemma. After applying the adherence-based algorithm, the agents will temporarily give up the environmental reward to clean up the river for the adherence reward, thus breaking the dilemma. At the beginning of training, the agents are not equal, but in the case of low resources, greedy agents will also tend to contribute. This eventually allows each agent to learn to clean the river and collect apples and achieve almost equality.

The Harvest game requires passive abstention rather than active provision. In this setting, adherence rewards provide signals for sustainable behavior so that the group can obtain more rewards. However, in the PPO method, because of the characteristics of the PPO algorithm, the magnitude of the gradient update is limited. In the early stage of training, the agents continuously explore the environment and learn that collecting apples can be rewarded, resulting in the faster and faster collection of the agent. After reaching a critical point, the speed of collection far exceeds the speed at which apples are regrowing, and the total number of apples collected becomes less and less. However, due to the nature of PPO, the agents can only learn to collect faster and faster, thus falling into a vicious circle. Our algorithm can only mitigate the rate of decline, but not solve the dilemma.

In general, the advantage of the adherence-based algorithm is the ability to maintain equality between agents on the basis of solving social dilemmas. However, the disadvantages are also obvious. In the cleanup game, under the premise of ensuring that the agents are not exploited, there will be problems of inefficiency. However, the practical significance of the adherence-based algorithm is more important.

6. Conclusions and Future Work

In multi-agent reinforcement learning to solve sequential social dilemmas, there is a lack of a practical method that can guarantee both the independence of agents and their equality. In this paper, we propose an adherence-based multi-agent reinforcement learning algorithm that improves the performance of RL agents by rewarding adherence to promote coordination and cooperation, which are lacking in sequential social dilemmas (SSDs). First, we defined systematically the evaluation of adherence based on counterfactual reasoning methods. Second, an adherence evaluation model was proposed to train agents' policies and learn to model other agents. Third, we design rewards for the agents to make them cooperate better. Finally, we utilize two SSDs environments to verify the effectiveness of our method. The advantage of the algorithm is that the independence and equality of agents can be guaranteed. However, it also comes with inefficiencies.

We conducted experiments in partially observable Markov games. The experimental results of total reward show that the adherence-based algorithm can effectively improve the collective reward. By comparing the results for baseline and the adherence-based algorithm, we verified that our method promotes mutual cooperation for every agent.

While the proposed algorithm verified its validity, there are still several directions worth studying. First, we plan to study other adherence evaluation approaches for the proposed method. Second, we use collective rewards to indirectly estimate the cooperation between agents. Based on this, the cooperative behavior between agents can be quantified, and we can train RL agents better by combining this metric with the reward. Third, the number of people is changing over time in the real world, which makes it necessary to take the dynamic aspects into account in future work.

Author Contributions: Conceptualization, Y.Y., T.G., P.Z. and H.J.; formal analysis, T.G.; investigation, T.G. and P.Z.; methodology, T.G. and H.J.; project administration, Y.Y.; resources, T.G.; software, T.G.; supervision, H.J.; validation, T.G., Y.Y. and P.Z.; visualization, T.G. and P.Z.; writing—original draft preparation, T.G.; writing—review and editing, Y.Y. and T.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (No. 91118002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Matignon, L.; Laurent, G.J.; Le Fort-Piat, N. Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems. *Knowl. Eng. Rev.* **2012**, *27*, 1–31. [CrossRef]
2. Wang, S.; Jia, D.; Weng, X. Deep reinforcement learning for autonomous driving. *arXiv* **2018**, arXiv:1811.11329.
3. Cobbe, K.; Hesse, C.; Hilton, J.; Schulman, J. Leveraging procedural generation to benchmark reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 2048–2056.
4. Yang, Y.; Caluwaerts, K.; Iscen, A.; Zhang, T.; Tan, J.; Sindhwani, V. Data efficient reinforcement learning for legged robots. In Proceedings of the Conference on Robot Learning, PMLR, Virtual, 16–18 November 2020; pp. 1–10.
5. Delaram, J.; Houshamand, M.; Ashtiani, F.; Valilai, O.F. A utility-based matching mechanism for stable and optimal resource allocation in cloud manufacturing platforms using deferred acceptance algorithm. *J. Manuf. Syst.* **2021**, *60*, 569–584. [CrossRef]
6. Yang, Y.; Hao, J.; Chen, G.; Tang, H.; Chen, Y.; Hu, Y.; Fan, C.; Wei, Z. Q-value path decomposition for deep multiagent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 10706–10715.
7. Bakolas, E.; Lee, Y. Decentralized game-theoretic control for dynamic task allocation problems for multi-agent systems. In Proceedings of the 2021 American Control Conference (ACC), New Orleans, LA, USA, 25–28 May 2021; pp. 3228–3233.
8. Lian, F.; Chakrabortty, A.; Duel-Hallen, A. Game-theoretic multi-agent control and network cost allocation under communication constraints. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 330–340. [CrossRef]
9. Huang, K.; Chen, X.; Yu, Z.; Yang, C.; Gui, W. Heterogeneous cooperative belief for social dilemma in multi-agent system. *Appl. Math. Comput.* **2018**, *320*, 572–579. [CrossRef]
10. Dobrowolski, Z. Internet of things and other e-solutions in supply chain management may generate threats in the energy sector—The quest for preventive measures. *Energies* **2021**, *14*, 5381. [CrossRef]
11. Leibo, J.Z.; Dueñez-Guzman, E.A.; Vezhnevets, A.; Agapiou, J.P.; Sunehag, P.; Koster, R.; Matyas, J.; Beattie, C.; Mordatch, I.; Graepel, T. Scalable evaluation of multi-agent reinforcement learning with melting pot. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 6187–6199.
12. Izquierdo, S.S.; Izquierdo, L.R.; Gotts, N.M. Reinforcement learning dynamics in social dilemmas. *J. Artif. Soc. Soc. Simul.* **2008**, *11*, 1.
13. Macy, M.W.; Flache, A. Learning dynamics in social dilemmas. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7229–7236. [CrossRef]
14. Van Lange, P.A.; Joireman, J.; Parks, C.D.; Van Dijk, E. The psychology of social dilemmas: A review. *Organ. Behav. Hum. Decis. Process.* **2013**, *120*, 125–141. [CrossRef]

15. Sandholm, T.W.; Crites, R.H. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* **1996**, *37*, 147–166. [CrossRef]
16. Sibly, H.; Tisdell, J. Cooperation and turn taking in finitely-repeated prisoners' dilemmas: An experimental analysis. *J. Econ. Psychol.* **2018**, *64*, 49–56. [CrossRef]
17. Busoniu, L.; Babuska, R.; De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2008**, *38*, 156–172. [CrossRef]
18. Leibo, J.Z.; Zambaldi, V.; Lanctot, M.; Marecki, J.; Graepel, T. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv* **2017**, arXiv:1702.03037.
19. Vinyals, O.; Ewalds, T.; Bartunov, S.; Georgiev, P.; Vezhnevets, A.S.; Yeo, M.; Makhzani, A.; Küttler, H.; Agapiou, J.; Schrittwieser, J.; et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv* **2017**, arXiv:1708.04782.
20. Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Dębiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. Dota 2 with large scale deep reinforcement learning. *arXiv* **2019**, arXiv:1912.06680.
21. Singh, S.; Barto, A.G.; Chentanez, N. *Intrinsically Motivated Reinforcement Learning*; Technical Report; Massachusetts University Amherst Dept of Computer Science: Amherst, MA, USA, 2005.
22. Eccles, T.; Hughes, E.; Kramár, J.; Wheelwright, S.; Leibo, J.Z. Learning reciprocity in complex sequential social dilemmas. *arXiv* **2019**, arXiv:1903.08082.
23. Chentanez, N.; Barto, A.; Singh, S. Intrinsically motivated reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2004**, *17*, 1281–1288.
24. Mohamed, S.; Jimenez Rezende, D. Variational information maximisation for intrinsically motivated reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2125–2133.
25. Klyubin, A.S.; Polani, D.; Nehaniv, C.L. Empowerment: A universal agent-centric measure of control. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–4 September 2005; Volume 1, pp. 128–135.
26. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-driven exploration by self-supervised prediction. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 16–17.
27. Schmidhuber, J. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Trans. Auton. Ment. Dev.* **2010**, *2*, 230–247. [CrossRef]
28. Peysakhovich, A.; Lerer, A. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv* **2017**, arXiv:1709.02865.
29. 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. Inequity aversion improves cooperation in intertemporal social dilemmas. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 3326–3336.
30. Wang, J.X.; Hughes, E.; Fernando, C.; Czarnecki, W.M.; Duéñez-Guzmán, E.A.; Leibo, J.Z. Evolving intrinsic motivations for altruistic behavior. *arXiv* **2018**, arXiv:1811.05931.
31. Jaques, N.; Lazaridou, A.; Hughes, E.; Gulcehre, C.; Ortega, P.; Strouse, D.; Leibo, J.Z.; De Freitas, N. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 3040–3049.
32. Yuan, Y.; Zhao, P.; Guo, T.; Jiang, H. Counterfactual-Based Action Evaluation Algorithm in Multi-Agent Reinforcement Learning. *Appl. Sci.* **2022**, *12*, 3439. [CrossRef]
33. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual multi-agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
34. Devlin, S.; Yliniemi, L.; Kudenko, D.; Tumer, K. Potential-based difference rewards for multiagent reinforcement learning. In Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems, Paris, France, 5–9 May 2014; pp. 165–172.
35. Shapley, L.S. Stochastic games. *Proc. Natl. Acad. Sci. USA* **1953**, *39*, 1095–1100. [CrossRef]
36. Littman, M.L. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*; Elsevier: Berlin/Heidelberg, Germany, 1994; pp. 157–163.
37. Kollock, P. Social dilemmas: The anatomy of cooperation. *Annu. Rev. Sociol.* **1998**, *24*, 183–214. [CrossRef]
38. Conybeare, J.A. Public goods, prisoners' dilemmas and the international political economy. *Int. Stud. Q.* **1984**, *28*, 5–22. [CrossRef]
39. Shankar, A.; Pavitt, C. Resource and public goods dilemmas: A new issue for communication research. *Rev. Commun.* **2002**, *2*, 251–272.
40. Hardin, G. The Tragedy of the Commons. *Science* **1968**, *162*, 124–1248. [CrossRef]
41. Dawes, R.M.; McTavish, J.; Shaklee, H. Behavior, communication, and assumptions about other people's behavior in a commons dilemma situation. *J. Personal. Soc. Psychol.* **1977**, *35*, 1. [CrossRef]
42. Perolat, J.; Leibo, J.Z.; Zambaldi, V.; Beattie, C.; Tuyls, K.; Graepel, T. A multi-agent reinforcement learning model of common-pool resource appropriation. *arXiv* **2017**, arXiv:1707.06600.
43. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1928–1937.

44. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef]
45. Ferguson, H.J.; Scheepers, C.; Sanford, A.J. Expectations in counterfactual and theory of mind reasoning. *Lang. Cogn. Process.* **2010**, *25*, 297–346. [CrossRef]
46. Tomasello, M. *Why We Cooperate*; MIT Press: Cambridge, MA, USA, 2009.
47. Oliver, P. Rewards and punishments as selective incentives for collective action: Theoretical investigations. *Am. J. Sociol.* **1980**, *85*, 1356–1375. [CrossRef]
48. O'Gorman, R.; Henrich, J.; Van Vugt, M. Constraining free riding in public goods games: Designated solitary punishers can sustain human cooperation. *Proc. R. Soc. B Biol. Sci.* **2009**, *276*, 323–329. [CrossRef] [PubMed]