applied sciences

MDPI

Article

Admission-Based Reinforcement-Learning Algorithm in Sequential Social Dilemmas

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

[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: Recently, the social dilemma problem is no longer limited to unrealistic stateless matrix games but has been extended to temporally and spatially extended Markov games by multi-agent reinforcement learning. Many multi-agent reinforcement-learning algorithms have been proposed to solve sequential social dilemmas. However, most current algorithms focus on cooperation to improve the overall reward while ignoring the equality among agents, which could be improved in terms of practicality. Here, we propose a novel admission-based hierarchical multi-agent reinforcement-learning algorithm to promote cooperation and equality among agents. We extend the give-or-take-some model to Markov games, decompose the fairness of each agent, and propose an Admission reward. For better learning, we design a hierarchy consisting of a high-level policy and multiple low-level policies, where the high-level policy maximizes the Admission reward by choosing different low-level policies to interact with environments. In addition, the learning and execution of policies are realized through a decentralized method. We conduct experiments in multiple sequential social dilemmas environments and show that the Admission algorithm significantly outperforms the baselines, demonstrating that our algorithm can learn cooperation and equality well.

Keywords: multi-agent reinforcement learning; hierarchical network; the give-or-take-some paradigm; sequential social dilemmas

check for updates

Citation: Guo, T.; Yuan, Y.; Zhao, P. Admission-Based Reinforcement-Learning Algorithm in Sequential Social Dilemmas. *Appl. Sci.* **2023**, *13*, 1807. https://doi.org/10.3390/app13031807

Academic Editors: Maxim Mozgovoy and Paolo Burelli

Received: 13 December 2022
Revised: 19 January 2023
Accepted: 28 January 2023
Published: 31 January 2023

1. Introduction

Reinforcement learning (RL) is a machine-learning method that improves the behavior of an agent by making it trial and error in an environment with given rules [1]. It is widely used in sequential decision problems to solve various tasks because of its theoretical generality [2]. Reinforcement learning has achieved numerous successes in many scenarios, such as video games [3], autonomous vehicles [4], traffic control [5], etc.

According to the number of controlled agents, reinforcement learning is divided into single-agent and multi-agent learning. However, most real-world scenarios involve multiple agents; therefore, single-agent reinforcement learning is insufficient. Multi-agent scenarios are more complex, and multi-agents need to consider environmental factors and interact with other agents during training [6,7]. Therefore, more researchers began to focus on multi-agent scenarios and multi-agent reinforcement learning (MARL).

One of the more famous scenarios is the social dilemmas, in which short-term individual incentives conflict with long-term collective interests [8,9]. In social dilemmas, a Nash equilibrium is not an ideal solution [10]. Humans, for example, face the dilemmas of collectively storing food in the summer in response to harsh winters, organizing annual irrigation system maintenance, or sustainably sharing local fisheries. In these cases, it is nearly impossible to use traditional models of human behavior based on a rational choice to guide cooperation [11,12]. Fehr and Falk proposed formal theories about fairness and reciprocity based on game theory respectively [13,14]. Nevertheless, these two models have a limited scope of applicability and can only work in matrix game social dilemmas (see, e.g., [15,16]). Ref. [17] proposed two more realistic video games, sequential social dilemmas (SSDs), such as those presented in behavioral studies [18,19]. In these environments, the agent does not simply choose cooperation or betrayal as atomic actions as in the matrix game. In addition, as the number of agents in a multi-agent system increases, the complexity of the problem grows exponentially. This poses a challenge for traditional reinforcement-learning methods, which need to be better suited to handle such complex environments.

A large body of work has investigated deep MARL methods from different perspectives to address these challenging tasks [20,21]. LOLA considers other agents' behavior strategies, adjusts its own strategy parameters, and finally successfully realizes complex multi-agent coordination [22]. Ref. [23] used inequality aversion to directly and indirectly change the payoff structure of agents by considering the rewards from other agents to facilitate agent cooperation in multiple SSD environments. However, both approaches make unrealistic assumptions about the accessibility of other agents' policy parameters or earned rewards. Ref. [24] injected different degrees of social value orientation into reinforcement learning agents, increasing the probability of group cooperation. In addition, using the social influence of each agent as an intrinsic incentive can also facilitate multi-agent cooperation [25]. However, the above two methods may cause some agents to be continuously exploited. A counterfactual-based contribution evaluation algorithm is proposed to give additional rewards by calculating the contribution of actions to latent states [26]. However, they adopted a centralized training method. Although the above algorithms have been successfully applied to multi-agent sequential dilemmas, most attempt to maximize the rewards between groups regardless of the situation in which some agents are exploited.

The concept of fairness plays a key role in both human society and multi-agent systems. Fairness is often seen as a critical factor in maintaining stability and increasing productivity [27,28]. Thus, we incorporate fairness into agent learning to prevent some agents from being exploited. However, simultaneously pursuing overall interests and fairness can create the problem of multi-objective conflicts in learning. To address this issue, a novel admission-based multi-agent reinforcement-learning algorithm, the Admission algorithm, is proposed, which enables agents to learn to be cooperative and fair.

First, we generalize the give-or-take-some dilemma (GOTS) [29] model to Markov games and introduce a fair Admission reward that allows each agent to optimize its own strategy. The individuals in the GOTS model take atomic actions, which makes it difficult for the agent to learn effectively in long-term tasks. To overcome this problem, we design an Admission Hierarchical Network (AHN) consisting of a high-level policy and several low-level policies. The high-level policy interacts directly with the environment and maximizes the Admission reward by choosing the low-level policy. Two low-level policies are specified to maximize the environmental reward and maximize contribution and the other strategies explore various behaviors guided by information-theoretic rewards. In addition, employing cheap talk [30], the algorithm coordinates agents' strategies in fully decentralized multi-agent learning. In this method, the agents are trained independently and execute independently. Therefore, the Admission algorithm avoids unrealistic assumptions and effectively solves the social dilemma problem.

We conducted some experiments in two classic SSD environments to evaluate the proposed algorithm. In the Admission algorithm, each agent directs behavior and enables simple communication through its own AHN. The experimental results show that the algorithm achieved cooperation among multiple agents and is superior to the existing algorithms in fairness. Due to cheap talk, the Admission algorithm can learn and execute in a completely decentralized manner, making it more realistic.

The rest of this paper is organized as follows: In Section 2, we provide an overview of the partial observation Markov decision process, hierarchical reinforcement learning, and sequential social dilemmas. In Section 3, we describe the proposed Admission algorithm in detail, including its reward design, network structure, and algorithm flow. Section 4 provides the experimental environment and results, while Section 5 offers a discussion of these findings. Finally, in Section 6, we summarize the paper.

2. Backgrounds

Our study is about solving sequential social dilemmas with hierarchical RL. Therefore, in this section, we review the basic Markov concepts, the principles of hierarchical reinforcement learning, and the relevant background on sequential social dilemmas.

2.1. Partially Observable Markov Decision Process

In this paper, we consider partially observable sequential decision-making games as a mathematical framework for studying MARL [31,32]. In this framework, agents collectively perform actions to interact with the environment on a discrete time scale. The environment then feeds individual rewards and partial observations of the state space to each agent. The rewards and observations determine what agents will do next, as agents need to learn to maximize individual rewards through their respective experiences. We formalize this framework as below.

A N-player partially observable sequential decision-making game defined over a set of states S can be mathematically modeled as a partially observable Markov decision process $\mathcal{M}$(POMDP). In $\mathcal{M}$, the observation function $\mathcal{O} : \mathcal{S} \times \{1, \ldots, N\} \rightarrow \mathbb{R}^d$ specifies each player's view on the state space. The observation space of player i is written as $\mathcal{O}^i = \{o^i | s \in \mathcal{S}, o^i = \mathcal{O}(s,i)\}$. At each timestep, the player samples action a^i from the action set $\mathcal{A}^1, \ldots, \mathcal{A}^N$, where $\mathcal{A}^i$ represents each player's action set. In the process of Markov state transition, the next state only depends on the current state and action—that is, $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \rightarrow \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ represents probability distributions over $\mathcal{S}$ after joint actions $a^1, \ldots, a^N \in \mathcal{A}^1, \ldots, \mathcal{A}^N$. The reward function $r^i : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \rightarrow \mathbb{R}$ defines the reward each player will receive in state $\mathcal{S}$.

Through their interactions with the environment, agents can learn their behavioral strategies $\pi^i : \mathcal{O}^i \rightarrow \Delta(\mathcal{A}^i)$ (written $\pi(a^i|o^i)$). For convenience, we simplify the joint action $(a^1, \ldots, a^N)$ and collective strategy $(\pi^1(\cdot|o^1), \ldots, \pi^N(\cdot|o^N))$ as $\vec{a}$ and $\vec{\pi}(\cdot|\vec{o})$. $\gamma \in [0,1]$ is the temporal discount factor, and the discount reward is $R_t = \sum_{l=0}^{\infty} \gamma^t r_{t+l}$. The objective function of each agent is to maximize its expected discount reward through its own policy, defined as:

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

2.2. Hierarchical Reinforcement Learning

Human decision-making often involves the choice of temporarily extended action routes over a wide range of time scales [33]. This can be challenging for a learning agent, as it requires the ability to operate at different levels of temporal abstraction. Ref. [34] introduced the concept of option and extended reinforcement learning to hierarchical reinforcement learning (HRL). In HRL, the agent occasionally chooses among a set of options, each of which defines a high-level policy that the agent can follow for a certain period. While following an option, the agent may need to take a sequence of primitive actions at each timestep in order to achieve the high-level goal defined by the option.

Compared with the RL algorithm, HRL decomposes the RL problem into multiple sub-problems, which have the following advantages [35]: higher interpretability, higher sample efficiency, and better solution to sparse reward problems. At the same time, HRL decomposes the complexity of policy learning into multiple levels, which significantly reduces the complexity of the problem and can well solve the problem of "*Curse of Dimensionality*". HRL thus succeeds in more complex and time-extended tasks. A well-known example is HDQN [36], where the meta-controller learns high-level policies to select goals, and the controller learns sub-policies to achieve goals.

This approach extends the traditional MDP to the semi-Markov decision process (SMDP) in order to model continuous-time discrete-event systems and learn high-level policies over multiple timesteps during which a given goal persists. In an SMDP, actions require a variable amount of time τ to simulate time-extended action processes. The transition function in this setting is defined as $\mathcal{T}(s_{t+\tau}|s_t, g_t)$, which denotes the probability of attaining goal g in state s_t, and reaching state $s_{t+\tau}$ after τ time steps. When state $s_{t+\tau}$ is reached, the agent receives the cumulative environmental reward $\tilde{r}_t$, where $\tilde{r}_t$ is the environmental reward. This modified transition function allows for the rewriting of the Bellman equation as follows:

$$V(s_t) = \tilde{r}_t(s_t, g_t) + \sum_{s_t,\tau} \gamma^\tau \mathcal{T}(s_{t+\tau}|s_t, g_t) V(s_{t+\tau}) \tag{2}$$

2.3. Sequential Social Dilemmas

The social dilemma problem provides a good scene for studying the tension between individual and collective rationality [12,37,38]. Social dilemmas have three characteristics [39]: (1) Individuals face some choices, some of which can maximize their own interests, while others are more beneficial to collective interests. (2) When everyone else chooses to maximize collective interests, the maximization of individual interests will bring greater benefits to individuals. (3) However, if all or enough people choose to maximize personal benefits, the outcome is much worse than if all people act to maximize collective benefits. Such as the famous "Tragedy of the Commons" [40]. The core of this dilemma is that individual rationality can lead to collective irrationality.

By the nature of the actors' tasks, the researchers differentiated between "Give-Some" (GS) and "Take-Some" (TS) social dilemmas [41]. In the GS dilemma, actors can keep private resources or use resources to develop or maintain a non-exclusive public good. For example, public broadcasting in the United States is available to everyone but depends on donations from a small number of institutions for funding. In this situation, cooperation means that individuals choose to contribute to the public good, while defection refers to the decision to withhold or not give. The TS dilemma and the GS dilemma are mirror images. In the TS dilemma, collective interests are competitive, i.e., one person's consumption reduces the amount enjoyed by others. For example, fishing. In this dilemma, actors are considered cooperative when they refrain from overconsumption and defectors when they do not.

Sequential social dilemmas(SSDs) are models proposed to simulate better real-world social dilemmas [17]. In SSDs, the decision-making process is modeled as a partially observable Markov game, where the actions and observations of individual agents are temporally and spatially extended. The underlying incentive structure of SSDs often leads to a reward-inhibition equilibrium in which individual agents are motivated to prioritize their short-term rewards over the collective good. Therefore, the sum of the rewards obtained by all agents becomes an explicit measure to estimate how well agents learn to cooperate [23]. In Section 4, two SSD environments in which traditional RL agents have difficulty learning to cooperate are described.

3. Method

In this section, we describe the Admission algorithm in detail. Section 3.1 outlines how the Admission reward is calculated. Section 3.2 details the Admission Hierarchical Network (AHN), and Section 3.3 describes the decentralized workflow of the Admission algorithm.

3.1. Admission Reward

The give-or-take-some (GOTS) paradigm, proposed in [29], introduces a more realistic hybrid social dilemma by allowing participants to both contribute and request resources from a shared pool. Considering there are N players, each of these players i has a endowment, e_i, (where $e_i \geq 0$ *and* $i = 1, \ldots, n$). Each player i can "contribute" $c_i \geq 0$, and/or "request" $r_i \geq 0$. The individual payoff is formulated as:

$$P_i = (e_i - c_i) + \zeta(r_i + b),\ where\ \zeta = 1\ if\ T \geq 0\ and\ \zeta = 0\ if\ T < 0, \tag{3}$$

where b is a positive bonus, and it depends on the sum of net contributions, $T = \sum(c_i - r_i)$, which we also call the admission boundary.

In real dilemmas, the contributed and demanded resources are often not the same resource, and there is no bonus. To this end, we set r_i as the environmental reward obtained by the player i, and then the utility of each player can be simplified as:

$$U_i = \zeta r_i,\ where\ \zeta = 1\ if\ T \geq 0\ and\ \zeta = 0\ if\ T < 0, \tag{4}$$

in which $T = \sum(\kappa c_i - r_i)$. The parameter κ controls the proportion of contribution and environmental reward, which depends on the specific environment.

However, as a computational model, the GOTS model is only suitable for matrix games. Equation (4) can only work in stateless games [42]. Thus, we need to extend this model to temporally extended Markov games. The main issue with redefining Equation (4) for a Markov game is that different players' rewards are obtained at different timesteps. Therefore, the key to extending Equation (4) to this case is to make each player's reward trajectory temporally smooth. Let $r_i(s, a)$ represent the reward the i-th player receives for taking action a in state s at time step t. We define the i-th player's Admission reward $u_i(s, a)$ and admission boundary at timestep t as:

$$\begin{cases} u_i(s_t^i, a_t^i) = \zeta r_i(s_t^i, a_t^i),\ where\ \zeta = 1\ if\ T \geq 0\ else\ \zeta = 0 \\ \qquad T = \sum[\kappa d_t^i(s_t^i, a_t^i) - e_t^i(s_t^i, a_t^i)] \end{cases} \tag{5}$$

where the temporal smoothed contributions d_t^i and rewards e_t^i of each agent i are updated at each timestep according to:

$$d_t^i(s_t^i, a_t^i) = \gamma\lambda d_{t-1}^i(s_{t-1}^i, a_{t-1}^i) + c_t^i(s_t^i, a_t^i) \tag{6}$$

$$e_t^i(s_t^i, a_t^i) = \gamma\lambda e_{t-1}^i(s_{t-1}^i, a_{t-1}^i) + r_t^i(s_t^i, a_t^i), \tag{7}$$

where λ is a hyperparameter.

In the sequential decision-making process of multi-agents, it is difficult for an individual agent to optimize the fairness of the whole because the fairness is not only related to its strategy but also the strategies of other agents. Then, due to the limited resources, cooperative agents are easily exploited by selfish agents. We decompose the fairness goal of each agent to optimize the Admission reward (5) further:

$$u_i(s_t^i, a_t^i) = \zeta \left(\frac{\epsilon + \bar{e}_{t-1}^i}{\epsilon + e_{t-1}^i} \right)^2 r_i(s_t^i, a_t^i), \tag{8}$$

where $\bar{e}$ is the mean value, ϵ is a small positive number to avoid a zero numerator or denominator. Rewards are optimized so that each agent responds to the actions of other agents. Therefore, Admission rewards can coordinate the policies of agents. At the same time, $\bar{e}$ can be calculated through a decentralized method.

3.2. Admission Hierarchical Network

In the GOTS model, individuals take atomic actions; however, in long-term sequences, it is difficult for the agent to maintain the initial behavior. For example, when an agent performs a contribution operation, and resources appear in its observations, the agent is likely to give up performing contributions and turn to collecting resources, making it challenging to learn contribution behaviors. Therefore, this is difficult for traditional reinforcement learning.

In this work, we design a two-level hierarchical network, Admission Hierarchical Network(AHN), for each agent to solve this problem. As shown in Figure 1, the AHN consists of a high-level policy and several low-level policies, respectively, denoted as π_{hi} and π_i. The $\pi_{hi}(z|o)$ selects a π_i based on partial observations of the environment, where $z \in Z$ is the space of possible low-level policies. The selected low-level policy starts to perform specific actions and obtains the corresponding reward r, while π_{hi} obtains Admission rewards u_t.

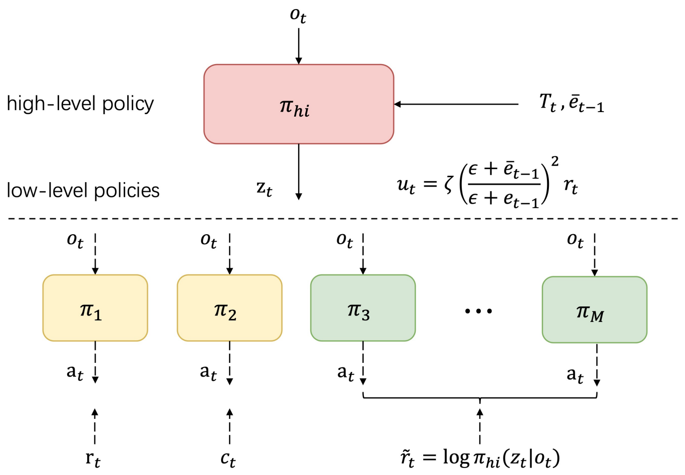

Figure 1. AHN architecture. The high-level policy π_{hi} selects a low-level policy π_i through observation, and then π_i performs actions to obtain rewards, while π_{hi} obtains Admission rewards u_t.

Unlike earlier work, the high-level policy does not switch between low-level policies every τ timestep. Instead, it executes the selected low-level policy at every timestep, such as the low-level policy. This is necessary because we have no way to ensure that the timesteps required by each strategy are the same. In addition, in order to avoid frequent replacement of low-level policies by high-level policy, the high-level policy will be subject to a minor penalty when each low-level policy is changed. This structure is illustrated in Figure 2.

To improve efficiency, we specify the low-level policy π_1 and π_2 to maximize the environmental reward r and the contribution reward, respectively. For other low-level policies, since we cannot directly quantify the differences in low-level policies, we utilize information theory to guide the exploration of various possible behaviors. In particular, we note that the greater the difference between the low-level policies, the less the uncertainty of Z given the observation, i.e., the lower the entropy of Z. From the high-level policy point of view, the low-level policies should be differentiated from each other in order to provide more choices. Therefore, we set the minimization of $H(Z|O)$ as the objective function of the low-level policies.

High-level policy only output probability distribution over the space of low-level policies $p_{hi}(z|o)$, and thus the goal of each low-level policy is to maximize the probability of being chosen by the high-level policy. To maximize the probability of being selected, we set the reward of the low-level policy to be $\log p_{hi}(z|o)$. On the other hand, in order to explore as many behaviors as possible, the low-level policies should take actions more randomly, i.e., we should maximize the entropy between observations and actions, $H(A|O)$. Therefore, we consider the term as an entropy regularization and place it into the objective function of policy learning. Policy learning with entropy regularization can be written as:

$$\begin{aligned} \max[J(\pi_i)] &= \max[H(A|O) - H(Z|O)] \\ &= \max[\mathbb{E}_{a\sim\pi_i}[-p(a|o)\log p(a|o)] + \mathbb{E}_{z\sim\pi_i, o\sim\pi_{hi}}[\log p(z|o)]. \end{aligned} \tag{9}$$

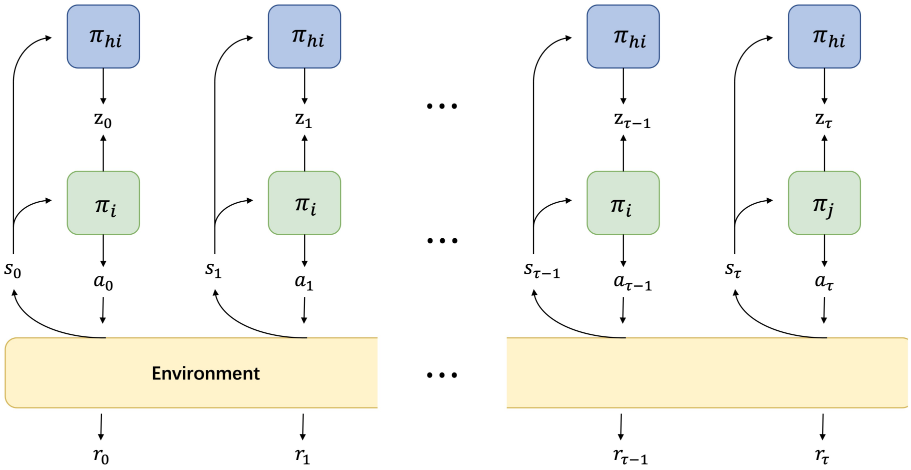

Figure 2. Unlike existing work, in the Admission algorithm, the high-level policy π_{hi} must make a choice every timestep. When the high-level policy π_{hi} changes its choice, it receives a small negative reward.

3.3. Decentralized Training

Centralized training is advantageous in coordinating the behaviors of all agents. However, the difficulty of centralized training increases exponentially as the number of agents grows. When the number of agents is large, the centralized training method may face the problem of dimensionality explosion. Our method is, thus, decentralized in both training and execution. Each agent does not need to access other agents, and we achieve cooperation by coordinating the agents through cheap talk [43,44].

During the decentralized training process, each agent needs to know the total contribution and total benefit of all agents to know its current admission boundary. When there are not many agents, obtaining contributions and rewards and computing admission boundaries from each agent are relatively straightforward. However, with a large number of agents, the cost of collecting data from all agents can become prohibitively expensive in practical applications. To solve this problem, we use cheap talk to calculate the admission boundary—that is, only to collect the information of neighboring agents to calculate their admission boundary. Let $\mathcal{N}_i$ be the set of adjacent agents observed by agent i. Each agent maintains an admission boundary and updates it by

$$T^i_{t+1} = \sum_{j\in\mathcal{N}_i} \kappa d^j_t(s^j_t, a^j_t) - e^j_t(s^j_t, a^j_t). \tag{10}$$

The cheap talk process is performed in a decentralized manner and requires only limited communication between neighboring agents to complete the evaluation.

Through cheap talk, let us calculate the boundaries. In this setting, a greedy agent needs to stay close to the contributing agent if it wants to free-ride. However, only a small reward can be obtained due to fairness factors. A selfless agent will be rewarded higher for fairness. This constraint also has some side effects that may bring agents closer together. However, this encouragement to draw closer is justified because people seek to belong and spend time near other people [45].

The training method is introduced in detail in Algorithm 1. Both high-level and low-level policies in our algorithm are trained using A3C [46] and PPO [47]. The high-level policy selects a low-level policy to take the original action at each step. When a low-level policy is replaced, it is updated based on the trajectory generated over its duration. The high-level policy is also updated based on the trajectories and rewards generated by the low-level policies during each episode.

Algorithm 1 Admission algorithm training.

We used two reinforcement-learning algorithms, A3C and PPO, which are referred to as *algo* below.

```
Initialize high-level policy and low-level policies for each agent $i$ with random parameters $\theta, \phi$. Initialize learning rate $\alpha$.
for each episodes do
    for $t = 1, \ldots,$max-episode-length do
        Calculate admission boundary $T_t^i$
        The high-level policy $\pi_\theta$ chooses one low-level policy $\pi_{\phi_k}$
        The chosen policy $\pi_{\phi_k}$ acts to the environment
        and obtain the reward $\begin{cases} r_t & if\ z = 1, \\ c_t & if\ z = 2, \\ \log p_\theta(z|o_t) & else \end{cases}$
        The high-level policy $\pi_\theta$ obtains the reward $u_i(s_t, a_t)$
        Send $d_t^i$ and $e_t^i$ to the neighbor agents using cheap talk
        if $\pi_\theta$ changes the low-level policy, $\pi_{t-1}^{\phi_j} \neq \pi_t^{\phi_k}$ then
            $\pi_\theta$ obtains a little punishment
            Compute gradient $\nabla_{\phi_j}$
            Update policy and network $pi_{\phi_j}$ using algo
        end if
    end for
    Compute gradient $\nabla_\theta$
    Update policy and network $\pi_\theta$ using algo
end for
```

4. Experiment

To evaluate the effectiveness of the proposed Admission algorithm, we test the algorithm in two SSD environments (https://github.com/eugenevinitsky/sequential_social_dilemma_games, accessed on 10 November 2022), Cleanup and Harvest. First, the concrete rules of the two environments are described. Secondly, we introduce the general evaluation metrics. Finally, the experimental results are presented.

4.1. Environment

4.1.1. Cleanup

As shown in (Figure 3A), Cleanup is a multiplayer game within a 25 × 18 grid-world environment. The black part of the environment is an orchard, and the blue part is a river. Players are rewarded with one point for each apple (green square) they collect. Apples grow at a rate that depends on the density of waste in the river. The more waste there is, the slower it will be. At the same time, the waste in the river will continue to increase. When the waste in the river reaches a certain percentage, the apples stop regrowing.

Players can clean up close-range waste by cleaning the beam. Therefore, if players want more rewards, they need to clean up the river to maintain the growth of apples. Additionally, players can punish other players by paying a small price for using the punishment beam (reward-1), and the player who was hit obtains a reward minus 50. Punishment plays a crucial role in sequential social dilemmas, which can be used to deter free riders [48,49].

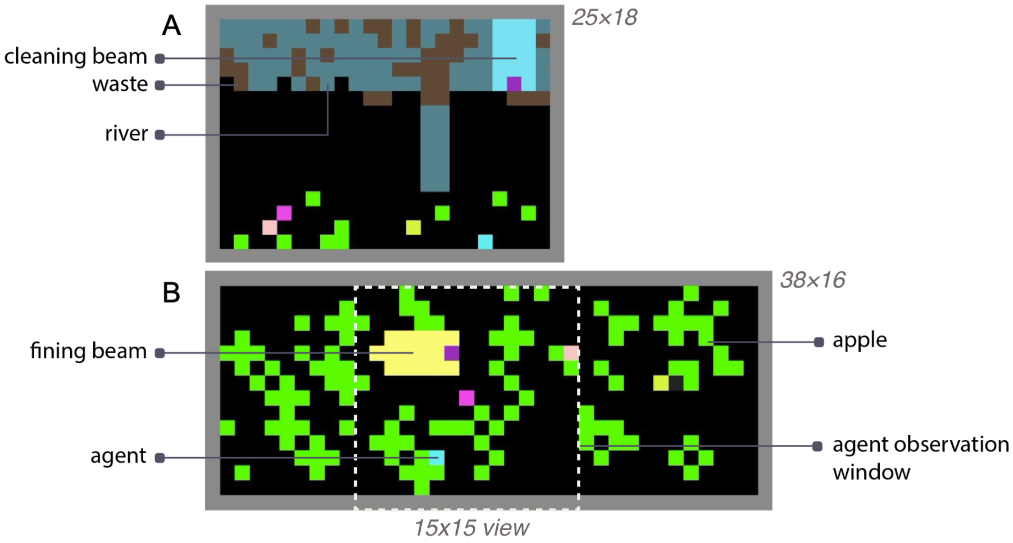

Figure 3. (**A**) is the Cleanup and (**B**) is the Harvest. Green grids represent apples, and the agent collects rewards by reaching green square locations. Blue and yellow represent cleaning beams and punishment beams, respectively.

The environment is initialized with no apples in the orchard and reaches the waste threshold in each episode. Apples will only grow if the player cleans up a portion of the waste. After 1000 time steps, the game ends, and the environment will be reset to its initial state. Each player's partially observable area is a 15 × 15 square centered on the player's current position.

In Cleanup, players earn rewards by collecting apples. Moreover in the case of not much waste, apples can be continuously regenerated. However, clearing the river does not directly benefit the player. Selfish players will simply choose to consume apples. In this environment, proactively cleaning the waterways is a behavior of cooperation, while not doing so or doing minimal cleaning is a behavior of defect. Therefore, in the Cleanup environment, we define cleanup as a contribution.

4.1.2. Harvest

The second experimental environment, the Harvest game, is shown in (Figure 3B). The Harvest game is a 24 × 26 grid world in which the player's goal is to collect as many apples as possible. The player is rewarded with a point for each apple collected. In Harvest, each grid's probability of apple regeneration depends on the number of nearby apples. The more apples around, the higher the probability of apple regeneration. The grid no longer grows apples when there are no apples around. Therefore, if players want more rewards, they need to not excessively collect to maintain the continuous growth of apples. In Harvest, the player also has the action of firing penalty beams. When a player uses punishment or is punished, the price he pays is the same as that of Cleanup.

Each episode in Harvest also has 1000 timesteps, and the environment is reset when the episode ends. Furthermore, the player's observability is limited to a 15 × 15 grid window centered on the player's current position. This means the player must use the information at their disposal to plan their actions and collect as many apples as possible within the time limit.

In Harvest, selfish players who excessively collect apples will ultimately cause the resources to be permanently depleted. The more greedy a player is, the more quickly the resources will be exhausted. Therefore, not collecting resources is a behavior of cooperation, while harvesting continuously is a behavior of defect. For this reason, in Harvest, we define not collecting apples as a contribution behavior.

4.2. Evaluation Metrics

In this paper, we evaluate our algorithm using three metrics [38]. The first metric, called the *Utilitarian* metric (U), is the sum of rewards obtained by all agents. The second metric, the *Equality* metric (E), measures equality between agents using the Gini coefficient. The third metric, the *Sustainability* metric (S), is the average timestep over which rewards are earned. These metrics allow us to evaluate the overall performance of our algorithm, as well as its ability to promote equality and sustainability.

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

4.3. Results

4.3.1. Cleanup

We evaluate the performance of agents in Cleanup using the collective reward, total waste cleaned, and equality of waste cleaned. We also evaluate an additional baseline using the A3C algorithm and PPO algorithm. The results can be seen in Figure 4.

The shades in the figure represent fluctuations in the experimental results of multiple groups. Experimental results show that in the early stages of training, the agent acts randomly to explore the environment. In the face of limited apples, agents start attacking each other, resulting in a sharp decline in total waste and equality. The agent then quickly learns not to attack; however, without the help of external forces, the baseline inevitably falls into a dilemma.

Last, the total reward in the A3C approach converges to around 20, and the total reward in the baseline-PPO slowly increases to 50. However, after applying the Admission algorithm, the collective rewards of the A3C and the PPO approaches quickly exceed the baseline and continue improving with the training time. After training for 30 million time steps, the algorithm (A3C) achieves a collective reward of around 350. The PPO approach reaches 400. At the same time, fairness in both A3C and PPO approaches is maintained at a high level.

As shown in Figure 4, in Cleanup, with the help of the admission-based algorithm, agents will obtain higher total rewards. Moreover, the baseline agent can only become stuck and unable to learn outcomes that are beneficial to the collective. From the properties of SSD, we can deduce that agents with high collective rewards learn to cooperate effectively. At the same time, the agents maintain relative equality under cooperation, proving that our algorithm can effectively solve SSDs.

4.3.2. Harvest

Figure 5 presents the performance of the baseline method and the admission-based algorithm in the Harvest game. We added a sustainability metric to evaluate the agents' behavior compared to the previous section.

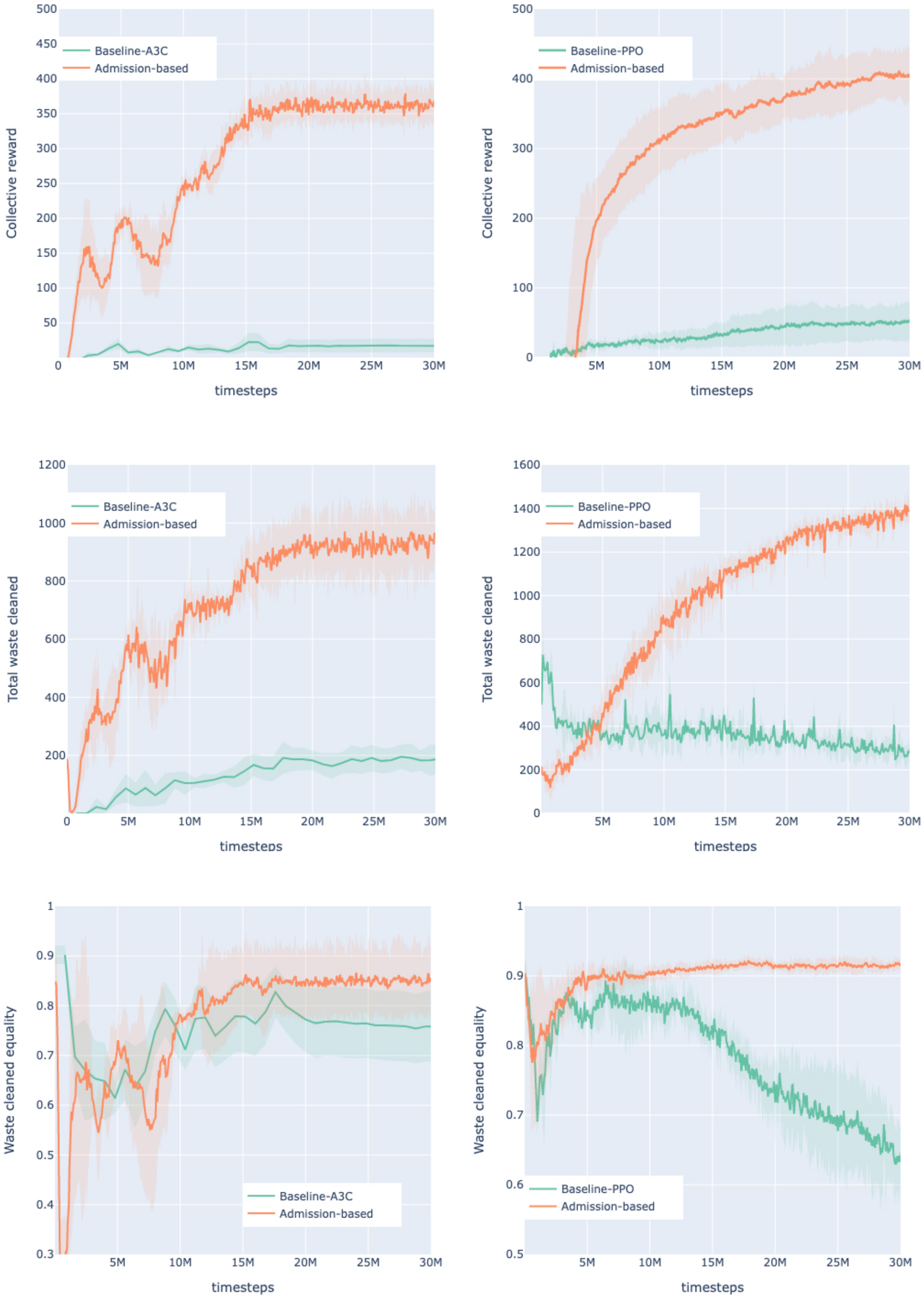

Figure 4. The performance of the Admission algorithm in various metrics in Cleanup. The three images on the left and the three images on the right are the results of our algorithm applied to A3C and PPO, respectively.

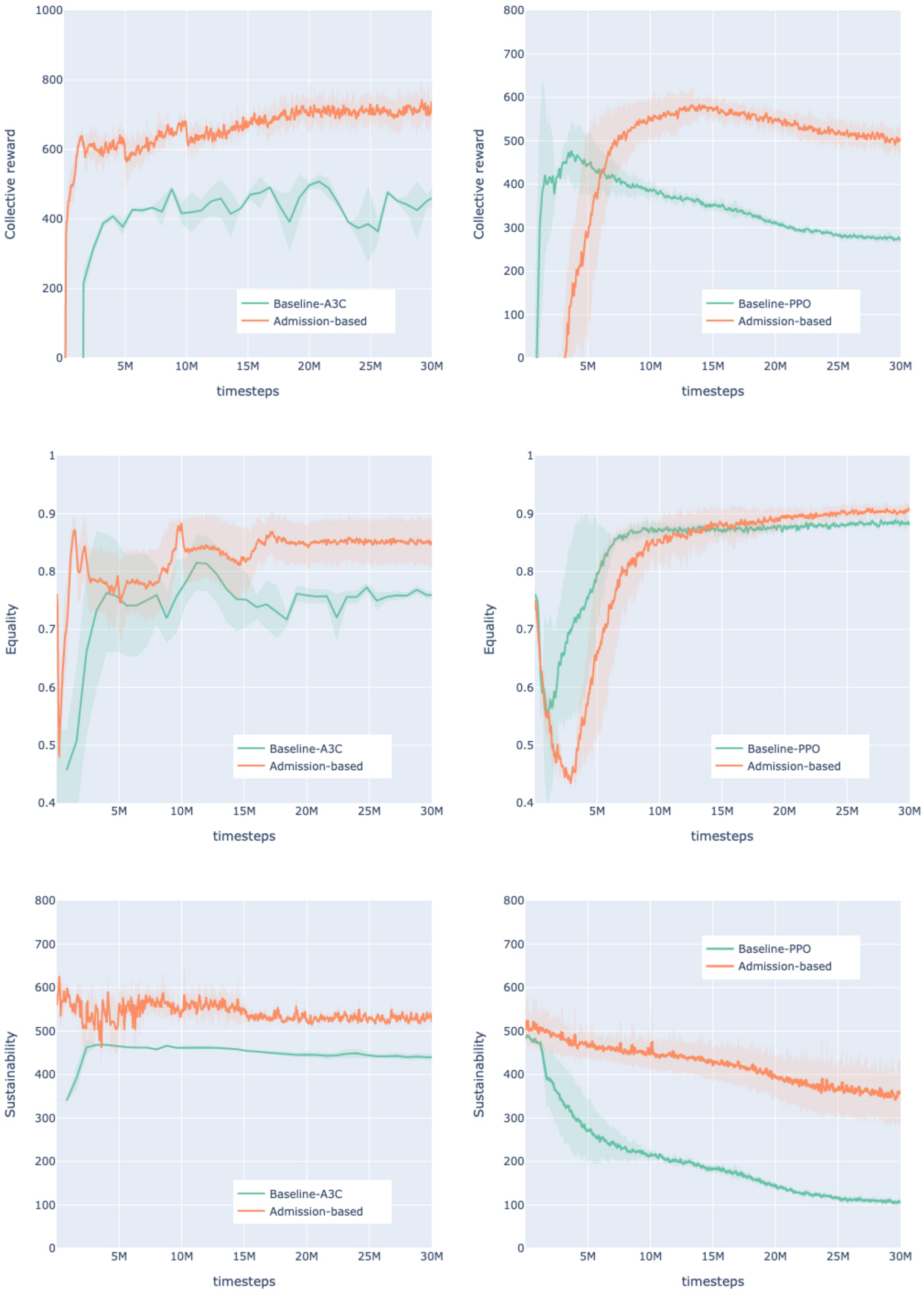

Figure 5. The performance of the Admission algorithm in various metrics in Harvest. The three images on the left and the three images on the right are the results of our algorithm applied to A3C and PPO, respectively. One of the metrics was replaced by the Sustainability metric.

At the beginning of training, the agents start exploring different places in the environment and start learning to collect apples. They also learn that their punitive action is ineffective and stop using it. Over time, the agents improve their ability to collect apples, and their collective rewards increase. The Baseline-A3C reaches a reward of 400, while the Baseline-PPO reaches around 450 but then starts to decline gradually. At the same time, the sustainability of baseline-PPO is also declining simultaneously. When the Admission algorithm is applied, the total rewards for the A3C approaches can converge to about 700, and the PPO approaches can reach up to about 600. Moreover, the fairness of both approaches is also guaranteed.

Experiments show that agents applying the Admission algorithm obtained more rewards. The density of apples determines the growth probability of an apple in the Harvest environment in its surrounding area. This suggests that maintaining a moderate number of apples may increase the overall rewards. According to this property, the improvement of collective reward reflects the success of our algorithm in breaking the social dilemma problem.

5. Discussion

In Cleanup, the agents are in a social dilemma because they are too greedy to clean up the waste. After applying the Admission algorithm, the agent will no longer greedily pursue environmental rewards because of the admission boundary and will clean up the waste for the Admission rewards, thereby, breaking the predicament. In the pre-training period, the agents are unequal. Nevertheless, as the training progresses, the agents tend to contribute when resources are low. Eventually, each agent learns to cooperate equally. In addition, the equality of Baseline-A3C is also high because the group is in a dilemma, and the individual rewards are meager. In the baseline-PPO, the agents learn simple cooperation; however, as the total reward slowly increases, the equality begins to decline.

In the context of the harvesting game, making a contribution does not involve actively paying but rather refraining from collecting resources. The Admission reward in the Admission algorithm incentivizes sustainable behavior and ultimately leads to higher collective rewards. However, the PPO approach has limitations in the distance of gradient updates due to the nature of the algorithm. This leads to agents exploring the environment and learning to obtain rewards more quickly at the beginning of training.

As the threshold is reached, the resource collection rate exceeds the resource regeneration rate, decreasing the total number of resources collected. Due to the constraints of the PPO approach, agents must collect resources at an increasingly fast rate, leading to a vicious cycle. As a result, the total reward and sustainability of the baseline-PPO approach initially increases but ultimately decreases. The Admission algorithm (PPO) can only mitigate this decline but cannot reverse the trend.

In general, the Admission algorithm has the advantage of being able to solve social dilemmas while maintaining equality among agents. However, the Admission algorithm also has disadvantages. In the Cleanup game, there is inefficiency. Nevertheless, the practical significance of the Admission algorithm is more important.

6. Conclusions

This paper presented an Admission-based hierarchical multi-agent reinforcement learning approach for tackling cooperation and equality in social dilemmas. The proposed algorithm extends the give-or-take-some model to Markov games and incorporates an Admission reward to facilitate the learning of cooperation and fairness among agents. The algorithm also includes an Admission Hierarchical Network (AHN), which comprises a high-level policy and multiple low-level policies. The high-level policy was trained to optimize the Admission reward, while the low-level policies were trained to optimize the environment reward and contribution reward. Additionally, other strategies were employed to provide diverse behaviors guided by the derived information-theoretic reward.

The use of cheap talk allows for decentralized learning and execution of the Admission algorithm. Through experiments on two SSD environments, we showed that the Admission algorithm effectively facilitated cooperation and fairness and outperformed the existing baselines in various scenarios.

Author Contributions: Conceptualization, T.G. and Y.Y.; Methodology, T.G.; Software, Y.Y.; Validation, T.G. and P.Z.; Formal analysis, P.Z.; Investigation, T.G.; Data curation, P.Z.; Writing—original draft, T.G.; Writing—review & editing, T.G.; Visualization, T.G.; Supervision, Y.Y.; Project administration, Y.Y.; Funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded 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. Kirk, R.; Zhang, A.; Grefenstette, E.; Rocktäschel, T. A survey of generalisation in deep reinforcement learning. *arXiv* **2021**, arXiv:2111.09794.
2. 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]
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, Virtual Event, 13–18 July 2020; pp. 2048–2056.
4. Aradi, S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 740–759. [CrossRef]
5. Singh, A.J.; Kumar, A.; Lau, H.C. Hierarchical multiagent reinforcement learning for maritime traffic management. In Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, 9–13 May 2020.
6. Zhang, H.; Feng, S.; Liu, C.; Ding, Y.; Zhu, Y.; Zhou, Z.; Zhang, W.; Yu, Y.; Jin, H.; Li, Z. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3620–3624.
7. Terry, J.; Black, B.; Grammel, N.; Jayakumar, M.; Hari, A.; Sullivan, R.; Santos, L.S.; Dieffendahl, C.; Horsch, C.; Perez-Vicente, R.; et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15032–15043.
8. Izquierdo, S.S.; Izquierdo, L.R.; Gotts, N.M. Reinforcement learning dynamics in social dilemmas. *J. Artif. Soc. Soc. Simul.* **2008**, *11*, 1.
9. Macy, M.W.; Flache, A. Learning dynamics in social dilemmas. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7229–7236. [CrossRef]
10. 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]
11. Feeny, D.; Berkes, F.; McCay, B.J.; Acheson, J.M. The tragedy of the commons: twenty-two years later. *Hum. Ecol.* **1990**, *18*, 1–19. [CrossRef]
12. Shankar, A.; Pavitt, C. Resource and public goods dilemmas: A new issue for communication research. *Rev. Commun.* **2002**, *2*, 251–272.
13. Fehr, E.; Schmidt, K.M. A theory of fairness, competition, and cooperation. *Q. J. Econ.* **1999**, *114*, 817–868. [CrossRef]
14. Falk, A.; Fischbacher, U. A theory of reciprocity. *Games Econ. Behav.* **2006**, *54*, 293–315. [CrossRef]
15. Sandholm, T.W.; Crites, R.H. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* **1996**, *37*, 147–166. [CrossRef]
16. de Cote, E.M.; Lazaric, A.; Restelli, M. Learning to cooperate in multi-agent social dilemmas. In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, Hakodate, Japan, 8–12 May 2006; pp. 783–785.
17. Leibo, J.Z.; Zambaldi, V.; Lanctot, M.; Marecki, J.; Graepel, T. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv* **2017**, arXiv:1702.03037.
18. Janssen, M.A.; Holahan, R.; Lee, A.; Ostrom, E. Lab experiments for the study of social-ecological systems. *Science* **2010**, *328*, 613–617. [CrossRef]
19. Janssen, M.A. The role of information in governing the commons: Experimental results. *Ecol. Soc.* **2013**, *18*, 4. [CrossRef]
20. 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.
21. Eccles, T.; Hughes, E.; Kramár, J.; Wheelwright, S.; Leibo, J.Z. Learning reciprocity in complex sequential social dilemmas. *arXiv* **2019**, arXiv:1903.08082.
22. Foerster, J.N.; Chen, R.Y.; Al-Shedivat, M.; Whiteson, S.; Abbeel, P.; Mordatch, I. Learning with opponent-learning awareness. *arXiv* **2017**, arXiv:1709.04326.

23. 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, Montréal, QC, Canada, 3–8 December 2018; pp. 3326–3336.
24. Peysakhovich, A.; Lerer, A. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv* **2017**, arXiv:1709.02865.
25. 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.
26. Yuan, Y.; Zhao, P.; Guo, T.; Jiang, H. Counterfactual-Based Action Evaluation Algorithm in Multi-Agent Reinforcement Learning. *Appl. Sci.* **2022**, *12*, 3439. [CrossRef]
27. Jiang, J.; Dun, C.; Huang, T.; Lu, Z. Graph convolutional reinforcement learning. *arXiv* **2018**, arXiv:1810.09202.
28. Chu, T.; Wang, J.; Codecà, L.; Li, Z. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1086–1095. [CrossRef]
29. McCarter, M.W.; Budescu, D.V.; Scheffran, J. The give-or-take-some dilemma: An empirical investigation of a hybrid social dilemma. *Organ. Behav. Hum. Decis. Process.* **2011**, *116*, 83–95. [CrossRef]
30. Foerster, J.; Assael, I.A.; De Freitas, N.; Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
31. Shapley, L.S. Stochastic games. *Proc. Natl. Acad. Sci. USA* **1953**, *39*, 1095–1100. [CrossRef]
32. Littman, M.L. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*; Elsevier: Amsterdam, The Netherlands, 1994; pp. 157–163.
33. Pateria, S.; Subagdja, B.; Tan, A.h.; Quek, C. Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [CrossRef]
34. Sutton, R.S.; Precup, D.; Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.* **1999**, *112*, 181–211. [CrossRef]
35. Al-Emran, M. Hierarchical reinforcement learning: A survey. *Int. J. Comput. Digit. Syst.* **2015**, *4*, 172–221. IJCDS/040207. [CrossRef]
36. Kulkarni, T.D.; Narasimhan, K.; Saeedi, A.; Tenenbaum, J. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
37. Kollock, P. Social dilemmas: The anatomy of cooperation. *Annu. Rev. Sociol.* **1998**, *24*, 183–214. [CrossRef]
38. 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.
39. 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]
40. Hardin, G. TheTragedyoftheCommons. *Sciences* **1968**, *162*, 1243–1248. [CrossRef]
41. McDaniel, W.C.; Sistrunk, F. Management dilemmas and decisions: Impact of framing and anticipated responses. *J. Confl. Resolut.* **1991**, *35*, 21–42. [CrossRef]
42. de Jong, S.; Tuyls, K. Human-inspired computational fairness. *Auton. Agents -Multi-Agent Syst.* **2011**, *22*, 103–126. [CrossRef]
43. Cao, K.; Lazaridou, A.; Lanctot, M.; Leibo, J.Z.; Tuyls, K.; Clark, S. Emergent communication through negotiation. *arXiv* **2018**, arXiv:1804.03980.
44. Lazaridou, A.; Hermann, K.M.; Tuyls, K.; Clark, S. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv* **2018**, arXiv:1804.03984.
45. Tomasello, M. *Why We Cooperate*; MIT Press: Cambridge, MA, USA, 2009.
46. 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, New York, NY, USA, 20–22 June 2016; pp. 1928–1937.
47. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
48. Oliver, P. Rewards and punishments as selective incentives for collective action: theoretical investigations. *Am. J. Sociol.* **1980**, *85*, 1356–1375. [CrossRef]
49. 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. Biol. Sci.* **2009**, *276*, 323–329. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.