Intrinsic Action Tendency Consistency for Cooperative Multi-Agent Reinforcement Learning

Junkai Zhang[1,2], **Yifan Zhang**[1,3,4*], **Xi Sheryl Zhang**[1,3,4], **Yifan Zang**[1,2], **Jian Cheng**[1,3,4]
[1] Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences, Nanjing
[4]Nanjing Artificial Intelligence Research of AI
{zhangjunkai2021, xi.zhang, zangyifan2019}@ia.ac.cn, {yfzhang, jcheng}@nlpr.ia.ac.cn

Abstract

Efficient collaboration in the centralized training with decentralized execution (CTDE) paradigm remains a challenge in cooperative multi-agent systems. We identify divergent action tendencies among agents as a significant obstacle to CTDE's training efficiency, requiring a large number of training samples to achieve a unified consensus on agents' policies. This divergence stems from the lack of adequate team consensus-related guidance signals during credit assignments in CTDE. To address this, we propose Intrinsic Action Tendency Consistency, a novel approach for cooperative multi-agent reinforcement learning. It integrates intrinsic rewards, obtained through an action model, into a reward-additive CTDE (RA-CTDE) framework. We formulate an action model that enables surrounding agents to predict the central agent's action tendency. Leveraging these predictions, we compute a cooperative intrinsic reward that encourages agents to match their actions with their neighbors' predictions. We establish the equivalence between RA-CTDE and CTDE through theoretical analyses, demonstrating that CTDE's training process can be achieved using agents' individual targets. Building on this insight, we introduce a novel method to combine intrinsic rewards and CTDE. Extensive experiments on challenging tasks in SMAC and GRF benchmarks showcase the improved performance of our method.

(a) (b)

Figure 1: The illustration of the consistent action tendency. In (a) and (b), our agents' health value is lower than the enemies'. At this point, attacking either Enemy 1 or 2 simultaneously are the two best team policies. In (a), Agent 1 and Agent 2 attack enemies separately without agreeing on a team policy. On the contrary, agents in (b) achieve a consistent goal policy and agree to attack a common enemy. To reflect the policy consistency among agents, we propose the concept of *action tendency*. It reflects the policy distribution of agents toward different actions. We propose this *action tendency* notion to distinguish it from *policy*, which is usually the epsilon-greedy of Q functions only concerning the largest output in value-based approaches.

Introduction

Cooperative multi-agent reinforcement learning (MARL) algorithms have shown the great capacity and potential to solve various real-world multi-agent tasks, such as automatic vehicles control (Sallab et al. 2017; Zhou et al. 2020b), traffic intelligence (Cao et al. 2012; Mushtaq et al. 2023; Wiering et al. 2000), resource management (Motlaghzadeh et al. 2023; Sallab et al. 2017), game AI (Berner et al. 2019; Lin et al. 2023) and robot swarm control (Dahiya et al. 2023; Hüttenrauch, Šošić, and Neumann 2017). In a cooperative multi-agent system (MAS), every agent relies on their local observation to cooperate toward a team goal and the environment feedbacks a shared team reward. There exist two major challenges in cooperative MAS: partial observability and scalability. Partial observability refers to the fact that agents can only access their local observations, resulting in unstable environments. Scalability refers to the challenge that the joint spaces of states and actions increase exponentially as the number of agents grows. To tackle these issues, *Centralized Training with Decentralized Execution* (CTDE) paradigm is proposed (Sunehag et al. 2017), which allows agents to access the global state in the training stage and take actions individually. Given the CTDE paradigm, massive deep MARL methods have been proposed including VDN (Sunehag et al. 2017), QMIX (Rashid et al. 2020), QTRAN (Son et al. 2019), QPLEX (Wang et al. 2020b) and so forth. Their excellent performance can be attributed to the credit assignments, as rewards are critical as the most direct and fundamental instructional signals to drive behaviors (Silver et al. 2021; Zheng et al. 2021; Mguni et al. 2021).

However, it turns out that the sparse team rewards provided by many MAS environments cannot supply sufficient guidance for coordination behaviors, which results in inefficient training (Matignon, Laurent, and Le Fort-Piat 2012).

*Corresponding author

We analyze the QMIX training process and realize that numerous unsuccessful episodes are caused by the inconsistency of team policy goals among agents like Figure 1 (a). Among these episodes, each agent's action tendency is not unified to the same global policy. We argue that the reward in MAS is the most essential instructional signal to drive behaviors and ascribe agents' action tendency inconsistency to CTDE's lack of sufficient team consensus guidance signals.

An effective solution to this challenge is to add intrinsic rewards into the CTDE paradigm. There exist two major problems: *how to design an intrinsic reward to guide agents' unified action tendency* and *how to integrate the intrinsic rewards into the CTDE framework?* In MARL, there are plenty of works designing intrinsic rewards including curiosity-based incentives (Böhmer, Rashid, and Whiteson 2019; Hernandez-Leal, Kartal, and Taylor 2019; Iqbal and Sha 2019; Zhang et al. 2023), the mutual influence among agents (Chitnis et al. 2020; Jaques et al. 2019; Wang et al. 2019) and other specific designs (Strouse et al. 2018; Ma et al. 2022; Mguni et al. 2021; Du et al. 2019). However, most of them are designed to enhance exploration and employed in independent training ways, which suffer from unstable dynamics of environments. To ease the latter problem, EMC (Zheng et al. 2021) proposed a curiosity-driven intrinsic reward and incorporated it into the CTDE training paradigm. Yet it averages the calculated intrinsic rewards and directly adds them to the global team reward, which results in losing the diversity of the intrinsic reward's adjustment toward credit assignments for each agent.

In this work, we propose our novel Intrinsic Action Tendency Consistency for the cooperative multi-agent reinforcement learning method. We hope to design intrinsic rewards on the basis of CTDE, so as to achieve consistent team policy goals among agents in the training process. Specifically, we first propose an action model to predict the central agent's action tendency. We define our intrinsic reward as the surrounding agents' action tendency prediction error toward the central agents. It encourages the central agent to take actions matching the prediction of their neighbors. After that, we propose theoretical analyses on CTDE and convert it into an equivalent variant, RA-CTDE. To appropriately utilize N intrinsic rewards like IQL (Tan 1993) training paradigm, we equivalently transform the original global target of CTDE into N ones. At last, we incorporate our action model based intrinsic reward into RA-CTDE and denote it by IAM. We integrate our method into QMIX and VDN, and conduct extensive experiments in StarCraft II Micromanagement environment (Samvelyan et al. 2019) (SMAC) and Google Football Research environment (Kurach et al. 2020) (GRF). Empirical results verify that our method achieves competitive performance and significantly outperforms other baselines.

Key contributions are summarized as follows: 1) We propose an action model based intrinsic reward measured by predicting the central agent's action tendency. 2) From a theoretical perspective, we address the issue of CTDE being unable to utilize the intrinsic rewards directly and consequently embed our intrinsic rewards into it. 3) By incorporating our method into QMIX and VDN, we demonstrate IAM's competitive performance on challenging MARL tasks.

Background

Dec-POMDP

A fully cooperative multi-agent task can be formulated as a *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP) (Oliehoek and Amato 2015), which is an augmented POMDP formulated by a tuple $\mathfrak{M} =< \mathcal{N}, \mathcal{S}, (\mathcal{O}_i)_{i\in\mathcal{N}}, (\mathcal{A}_i)_{i\in\mathcal{N}}, \mathbb{O}, \mathcal{P}, \mathcal{R}, \rho_0, \gamma >$, where every agent can only access the partial state of the environment and takes actions individually. Specifically, we denote $\mathcal{N} = \{1, ..., N\}$ as the set of agents, where N is the number of agents, $\mathcal{S}$ as the global finite state space, $\mathcal{O}_i$ as the partial observation of the state, obtained by the function $\mathbb{O}(s, i)|_{s\in\mathcal{S}}$, and $\mathcal{A}_i$ as the action space respectively. $\gamma \in [0, 1)$ is a discount factor and $\rho_0 : \mathcal{S} \to R$ is the distribution of the inital state s_0. The state transition probability function of the environment dynamics is $\mathcal{P} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$ where $\boldsymbol{\mathcal{A}} := \times_{i=1}^{N}\mathcal{A}_i$ is the joint action space selected by all agents. Due to the partial observable setting, every agent takes its observation-action history $\tau_i \in \{\mathcal{T}_i\}_{i=1}^N \equiv (\mathcal{O}_i \times \mathcal{A}_i)^* \times \mathcal{O}_i$ as the policy input to acquire more information. After agents taking their joint actions $\boldsymbol{a} : \{a_t^i\}_{i=1}^N$, the environment returns a team shared extrinsic reward r^{ext} by function $\mathcal{R}(\mathcal{S}, \boldsymbol{\mathcal{A}})$: $\mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathcal{R}$. We define the stochastic policy of agent i by $\pi_i(a_i|\tau_i) : \mathcal{T}_i \times \mathcal{A}_i \to [0, 1]$, the multi-agent system algorithms are designed to find optimal policies $\pi^* = \{\pi_i^*\}_{i=1}^N$ to maximize the joint extrinsic value function $V^{\boldsymbol{\pi}}(s) = \mathbb{E}_{s_0,\boldsymbol{a}_0,\dots}[\sum_{t=0}^{\infty}\gamma^t r_t^{ext}|\boldsymbol{\pi}]$, where $s_0 \sim \rho_0(s_0), \boldsymbol{\pi} = \{\pi_i\}_{i=1}^N$.

Centralized Training with Decentralized Execution

The primary challenge for MAS tasks is that agents can only access partial observation and are incapable to acquire the global state, to which an effective solution is the CTDE training paradigm (Bernstein et al. 2002). It allows all agents to access the global state in the centralized training stage and take actions individually in a decentralized manner. Formally, it formulates N individual Q-functions $\{Q_i(\tau_i, a_i; \theta_i)\}_{i\in\mathcal{N}}$ where θ_i is the network parameter for agent i. Meanwhile, it simultaneously preserves a joint action-value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$ constructed by individual Q functions to help training. In detail, the objective of CTDE is to get an optimal joint action-value function $Q_{tot}^*(s, \boldsymbol{a}) = r^{ext}(s, \boldsymbol{a}) + \gamma\mathbb{E}_{s'}[\max_{\boldsymbol{a'}} Q_{tot}^*(s', \boldsymbol{a'})]$. In the centralized training stage, Q-functions $\{Q_i\}_{i\in\mathcal{N}}$ are trained by minimizing the following target function:

$$\mathcal{L}^G(\boldsymbol{\theta}, \phi) = \mathbb{E}\left[r + \gamma\max_{\boldsymbol{a'}} Q_T(\boldsymbol{\tau'}, \boldsymbol{a'}) - Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta}, \phi)\right]^2 \quad (1)$$

$$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta}, \phi) = \mathcal{F}\left(Q_1(\tau_1, a_1), ..., Q_N(\tau_N, a_N), s; \phi\right) \quad (2)$$

where $\boldsymbol{\tau} = \{\tau_i\}_{i=1}^N, \boldsymbol{a} = \{a\}_{i=1}^N, \boldsymbol{\theta} = \{\theta_i\}_{i=1}^N$, ϕ is the parameters of the mixing network $\mathcal{F}$, and $\mathcal{D}$ is the replay buffer. $\{\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{\tau'}\} \sim \mathcal{D}$. Q_T denotes the expected return target function for the estimation of the global state-action pair. The gradients of $\boldsymbol{\theta}$ are calculated through function $\mathcal{F}$, which factorizes global Q_{tot} function into decentralized ones $\{Q_i\}_{i=1}^N$, motivating enormous efforts to find factorization structures among them (Sunehag et al. 2017; Rashid et al. 2020; Wang et al. 2020b).

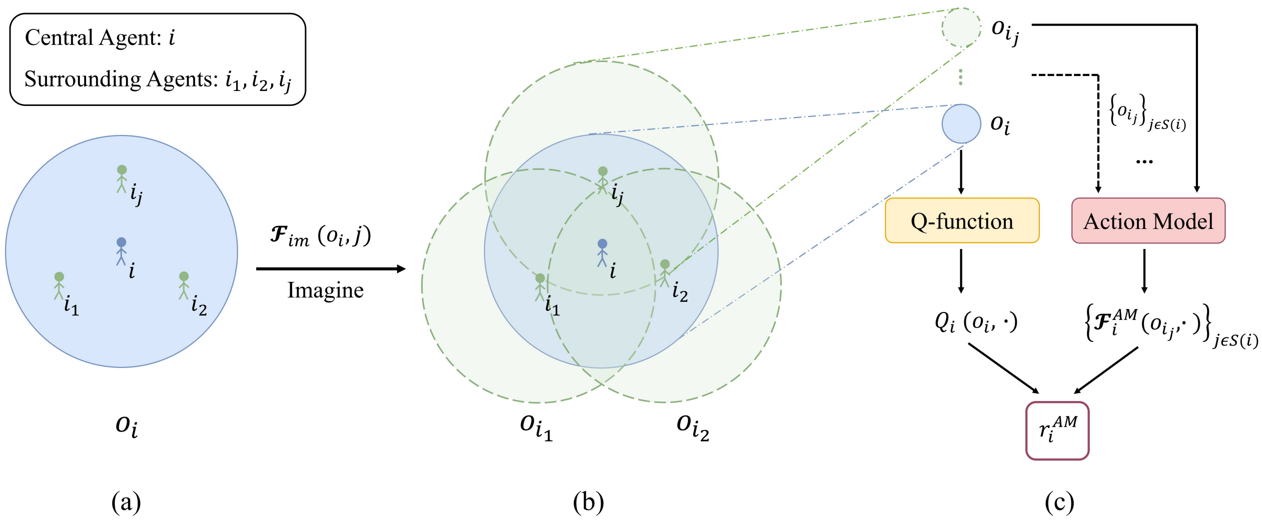


Figure 2: IAM-based reward. The blue and green zones represent the receptive field of the central agent and surrounding agents. The action model intrinsic reward is high when agent i takes actions that match their surrounding agents' predictions.

Related Works

Many of the intrinsic reward functions used in MARL have been adapted from single agent curiosity-based incentives (Hernandez-Leal, Kartal, and Taylor 2019; Iqbal and Sha 2019; Jaques et al. 2019), which aimed to encourage agents to explore their environment and seek out novel states. To better be applied in MARL, Some MARL-specific intrinsic reward functions have been proposed, including considering the mutual influence among agents (Chitnis et al. 2020; Jaques et al. 2019; Wang et al. 2019), encouraging agents to reveal or hide their intentions (Strouse et al. 2018) and predicting observation with alignment to their neighbors (Ma et al. 2022). Besides, Intrinsic rewards without task-oriented bias can increase the diversity of intrinsic reward space, which can be implemented by breaking the extrinsic rewards via credit assignment (Du et al. 2019) or using adaptive learners to obtain intrinsic rewards online (Mguni et al. 2021). Apart from independent manners to dealing with rewards, EMC (Zheng et al. 2021) proposed a curiosity-driven intrinsic reward and introduce an integration way to accomplish the CTDE training paradigm.

Method

In this section, we present our Intrinsic Action Tendency Consistency for cooperative MARL denoted by IAM (Intrinsic Action Model). Our purpose is to design an effective intrinsic reward to encourage consistent action tendencies and leverage it into CTDE in an appropriate manner. Specifically, we first introduce our action model based intrinsic reward, which encourages the central agent to take actions consistent with its neighbors' prospects. Then we propose a reward-additive equivalent variant of the CTDE framework denoted by RA-CTDE to incorporate our rewards reasonably. At last, we analyze the essential difference between VDN (Sunehag et al. 2017) and IQL (Tan 1993) and then demonstrate the reasonability of our reward integration way.

Action Model Based Intrinsic Reward

For a better interpretation, we first give the following definitions: As shown in Figure 2, when considering a specific Agent i, we define it as the central agent. Due to the partial observability of the environment, agents that are observable in the surrounding area of Agent i are defined as the surrounding agents and we denote the set as $S(i)$. During the training process, we hope that every central agent i will take into account its surrounding agents' expectations toward i's policy distribution. We denote its policy distribution by the action tendency, which represents the relative magnitude of an agent's inclination to take different actions.

Reward Calculation In a discrete action space, agent i's action tendency can be reflected from two different perspectives. From the viewpoint of the central agent i, its action tendency can be represented by its Q_i function. From the viewpoint of i's surrounding agents, we define the action models $\{\mathcal{F}_i^{AM}\}_{i=1}^N$ to allow them to predict the central agent's action tendency. $\mathcal{F}_i^{AM}$ is designed to utilize the same network structure as Q_i function. Their representation distance of action tendency reflects the central agent's consistency degree towards the surrounding agents' expectation. Therefore we formulate this distance as an action model based intrinsic reward, i.e. $\{r_i^{AM}\}_{i=1}^N$.

$$o_{i_j} = \mathcal{F}_{im}(o_i, j) \tag{3}$$

$$r_i^{AM} = \frac{-1}{|S(i)|} \sum_{j \in S(i)} Dis\big(\mathcal{F}_i^{AM}(o_{i_j}, \cdot\,; \omega_i) - Q_i(o_i, \cdot)\big) \tag{4}$$

The reward calculation process is illustrated in Figure 2 and Eq 3, 4. During the training phase, every agent first calculates its imagined surrounding agents' observations, then utilizes its Q function and action model to measure the action tendency distance, and finally obtains its action model based intrinsic reward r_i^{AM}. The imagine function

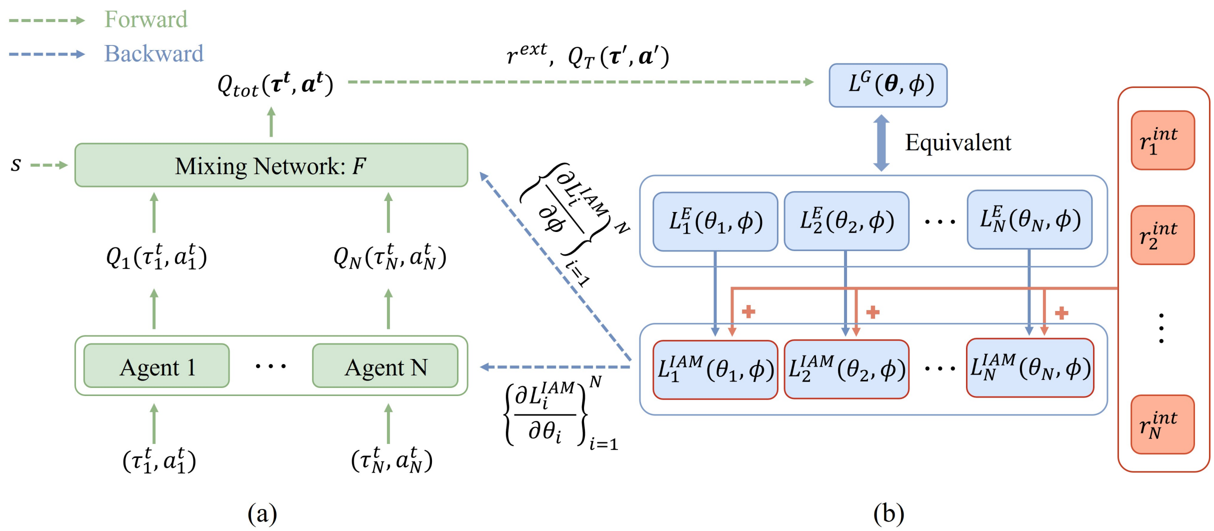


Figure 3: IAM training paradigm. The training paradigm consists of two stages: (a) Forward stage and (b) Backward stage. In the forward stage, we use mixing network $\mathcal{F}, Q_{tot}, r^{ext}$ and Q_T in Eq. 1 to calculate global TD-error target $\mathcal{L}^G$, which is the same as CTDE. In the backward stage, we first factorize $\mathcal{L}^G$ into N targets $\{\mathcal{L}_i^E\}_{i=1}^N$ by Eq 6 and 7, then add intrinsic rewards into them individually to obtain IAM targets: $\{\mathcal{L}_i^{IAM}\}_{i=1}^N$. The gradients of $\{\theta_i\}_{i=1}^N$ and ϕ are separately computed by backpropagating N targets $\{\mathcal{L}_i^{IAM}\}_{i=1}^N$.

$\mathcal{F}_{im}(o_i, j)$ in Eq 3l is defined to represent the surrounding agents' simulated observation *imagined* by the central agent i. The *imagining* process is realized by switching the viewpoints from the central agent into the surrounding agents, i.e., separately setting the positional coordinates of every surrounding agent as the origin to calculate the coordinates of the other agents attached with additional information, which does not require any learning parameters (more details in the Appendix). In experiments, we use the L_2 distance as the Dis function. Under this reward setting, agents are encouraged to take actions consistent with their surrounding agents' prospects.

$$
\begin{aligned}
Q_i^{(t+1)}(\tau_i,a_i) = & \underbrace{\mathop{\mathbb{E}}_{(\tau'_{-i},a'_{-i})\sim p_D(\cdot|\tau_i)}\left[y^{(t)}\left(\tau_i \oplus \tau'_{-i}, a_i \oplus a'_{-i}\right)\right]}_{\text{evaluation of the individual action } a_i} \\
& \underbrace{- \frac{n-1}{n} \mathop{\mathbb{E}}_{\boldsymbol{\tau}',\boldsymbol{a}'\sim p_D(\cdot|\Lambda^{-1}(\tau_i))}\left[y^{(t)}\left(\boldsymbol{\tau}', \boldsymbol{a}'\right)\right]}_{\text{counterfactual baseline}} \\
& + \underbrace{w_i\left(\tau_i\right)}_{\text{residue term}}
\end{aligned}
\tag{5}
$$

Action Model Training To obtain $\mathcal{F}^{AM}$, we use Q_i function values as supervised targets. This choice is reasonable based on the following insight: *The individual Q_i value incorporates interaction information of other agents to agent i, not just only the agent i's own action tendencies with linear value factorization.* In VDN training paradigm, the individual Q_i function can be factorized into Eq 5's form (Wang et al. 2020a), where $p_D(\cdot|\tau_i)$ denotes the conditional empirical probability of τ_i in the given dataset D , the notation $\tau_i \bigoplus \tau'_{-i}$ denotes $< \tau'_1, ..., \tau'_{i-1}, \tau_i, \tau'_{i+1}, ..., \tau'_n >$, and τ'_{-i} denotes the elements of all agents except for agent i.

In Eq 5, it is easy to see that the Q_i function essentially consists of three items, and the first two include the expectation of one-step TD target value over others. It indicates that the Q_i function value obtained in VDN includes the interactive historical expectation toward other agents. Although this analysis only applies to VDNs, we broaden the supervised target Q functions to QMIX and also achieve effective performance improvement. The pseudo-code of our algorithm is interpreted in the Appendix.

Reward-Additive CTDE (RA-CTDE)

The contradiction for CTDE to utilize N intrinsic rewards is that it has only one global target $\mathcal{L}^G$ during training. However, IQL (Tan 1993) can directly use N different intrinsic rewards naturally because it obtains N TD-losses individually. Based on that, we first factorize the global target $\mathcal{L}^G$ in Eq 1 into N individual ones and define it as Reward-Additive CTDE (RA-CTDE). Then we demonstrate its equivalence with the original target $\mathcal{L}^G$. At last, we discuss how to add intrinsic rewards to the RA-CTDE.

Definition 1. *(Reward-Additive CTDE). Let $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^N$ be the parameters of Q functions, $\mathcal{F}$ be the mixing network in CTDE, $\mathcal{N} = \{1, ..., N\}$ be the agents set, $\mathcal{Q}^N = \{Q_1(\tau_1, a_1; \theta_1), Q_2(\tau_2, a_2; \theta_2), ..., Q_N(\tau_N, a_N; \theta_N)\}, \{\boldsymbol{\tau}, \boldsymbol{a}, r^{ext}, \boldsymbol{\tau'}\} \sim \mathcal{D}$, assume $\forall i, j \in \mathcal{N}, \theta_i \neq \theta_j$, then Reward-Additive CTDE means computing $\{\mathcal{L}_i^E(\theta_i, \phi)\}_{i=1}^N$ in Eq 6 and Eq 7 and then updating their parameters respectively. The term $\mathcal{P}$ is not involved in the gradient calculation as a* *scalar. Formally:*

$$\mathcal{L}_i^E(\theta_i, \phi) = \mathbb{E}_{\boldsymbol{\tau}, \boldsymbol{a}, r^{ext}, \boldsymbol{\tau'} \in \mathcal{D}} \left[\mathcal{P} \cdot \mathcal{F}(\mathcal{Q}^N, s; \phi) \right] \quad (6)$$

$$\mathcal{P} = -2 \left(r^{ext} + \gamma \max_{\boldsymbol{a'}} Q_T(\boldsymbol{\tau'}, \boldsymbol{a'}) - \mathcal{F}(\mathcal{Q}^N, s; \phi) \right) \quad (7)$$

We propose our reward-additive variant of the CTDE framework RA-CTDE, where the $Q_T(\boldsymbol{\tau'}, \boldsymbol{a'})$ in Eq 7 is updated by the target function of the mixing network $\mathcal{F}$.[1] We consider that RA-CTDE is equivalent to the original CTDE paradigm based on the following theorem.

Theorem 1. *Let $\{\theta_i\}_{i=1}^N$ be the parameters of Q functions, ϕ be the parameters of the mixing network $\mathcal{F}$ in CTDE, $\mathcal{L}^G$ be the global target in Eq 1, $\mathcal{N} = \{1, ..., N\}$ be the agents set, $\mathcal{Q}^N = \{Q_1(\tau_1, a_1; \theta_1), Q_2(\tau_2, a_2; \theta_2), ..., Q_N(\tau_N, a_N; \theta_N)\}$, $\{\boldsymbol{\tau}, \boldsymbol{a}, r^{ext}, \boldsymbol{\tau'}\} \sim \mathcal{D}$, assume $\forall i, j \in \mathcal{N}, \theta_i \neq \theta_j$, then $\forall i \in \mathcal{N}$, the following equations hold:*

$$\frac{\partial \mathcal{L}^G(\boldsymbol{\theta}, \phi)}{\partial \theta_i} = \frac{\partial \mathcal{L}_i^E(\theta_i, \phi)}{\partial \theta_i} \quad (8)$$

$$\frac{\partial \mathcal{L}^G(\boldsymbol{\theta}, \phi)}{\partial \phi} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathcal{L}_i^E(\theta_i, \phi)}{\partial \phi} \quad (9)$$

The Theorem 1 is proved in the Appendix. According to it, we draw the conclusion that the CTDE's essence in updating gradients of $\{\theta_i\}_{i=1}^N$ and ϕ is to calculate the global target $\mathcal{L}^G$ and then respectively perform N gradient back-propagation steps for each agent. Therefore we can equivalently factorize the global target $\mathcal{L}^G$ into N individual ones denoted by $\mathcal{L}_i^E$ in RA-CTDE. The factorized target $\mathcal{L}_i^E$ provides an interface for adding rewards and we exhibit the reward-adding way based on the following corollary.

Corollary 1. *Let $\{\theta_i\}_{i=1}^N$ be the parameters of Q functions, $\mathcal{N} = \{1, ..., N\}$ be the agents set, $\mathcal{L}_i^{VDN}$ be the $\mathcal{L}_i^E$'s special case of VDN, $\{\boldsymbol{\tau}, \boldsymbol{a}, r^{ext}, \boldsymbol{\tau'}\} \sim \mathcal{D}$, assume $\forall i, j \in \mathcal{N}, \theta_i \neq \theta_j$, then $\forall i \in \mathcal{N}$:*

$$\mathcal{L}_i^{VDN}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau}, \boldsymbol{a}, r^{ext}, \boldsymbol{\tau'} \in \mathcal{D}} \left[\mathcal{P}_i \cdot Q_i(\tau_i, a_i; \theta_i) \right] \quad (10)$$

$$\mathcal{P}_i = -2 \left(r^{ext} + \mathcal{R}_i^{VDN} + \gamma \max_{a'} Q_i^-(\tau'_i, a'_i) - Q_i(\tau_i, a_i) \right) \quad (11)$$

$$\mathcal{R}_i^{VDN} = \gamma \max_{a'} \sum_{j=1, j \neq i}^{N} Q_j^-(\tau'_j, a'_j) - \sum_{j=1, j \neq i}^{N} Q_j(\tau_j, a_j) \quad (12)$$

We consider the special case VDN (Sunehag et al. 2017) and transform it into the RA-CTDE form, where the mixing network $\mathcal{F}$ is the calculation of summing over all Q_i functions. On the basis of the Eq 10, 11, and 12 in Corollary 1, we realize that VDN can be factorized into N targets like IQL(Tan 1993). But the essential difference between VDN and IQL (Tan 1993) is that the former adds certain intrinsic rewards $\{\mathcal{R}_i^{VDN}\}_{i=1}^N$ into $\{\mathcal{P}_i\}_{i=1}^N$. In other words, when the $\mathcal{R}_i^{VDN}$ are not incorporated in Eq 11, VDN fundamentally boils down to IQL. The reward $\mathcal{R}_i^{VDN}$ is incorporated into the TD-error term $\mathcal{P}_i$. We adopt the same reward-adding form as VDN and extend it to the RA-CTDE framework. Specifically, we choose to add the calculated intrinsic rewards with parameter β into N losses $\{\mathcal{L}_i^E\}_{i=1}^N$ and get our IAM targets in Eq 13, 14. The gradient of θ_i and ϕ can be obtained by computing $\frac{\partial \mathcal{L}_i^{IAM}(\theta_i, \phi)}{\partial \theta_i}$ and $\frac{1}{N} \cdot \sum_{i=1}^N \frac{\partial \mathcal{L}_i^{IAM}(\theta_i, \phi)}{\partial \phi}$ respectively. Figure 3 shows our whole training paradigm.

$$\mathcal{L}_i^{IAM}(\theta_i, \phi) = \mathbb{E}_{\boldsymbol{\tau}, \boldsymbol{a}, r^{ext}, \boldsymbol{\tau'} \in \mathcal{D}} \left[\mathcal{P}_i \cdot \mathcal{F}(\mathcal{Q}^N, s; \phi) \right] \quad (13)$$

$$\mathcal{P}_i = -2 \left(r^{ext} + \beta r_i^{int} + \gamma \max_{\boldsymbol{a'}} Q_T(\boldsymbol{\tau'}, \boldsymbol{a'}) - \mathcal{F}(\mathcal{Q}^N, s; \phi) \right) \quad (14)$$

Though the training paradigm of IAM also uses N targets motivated by the reward-adding way like IQL, the target $\mathcal{L}_i^{IAM}$ still contains other agents' information and the essence of IAM is an improved CTDE instead of an independent training method.

Experiments

To demonstrate the high efficiency of our algorithm, we exploit different environments to conduct a large number of experiments, including StarCraft II Micromanagement (SMAC) (Samvelyan et al. 2019), Google Research Football (GRF) (Kurach et al. 2020) and Multi-Agent Particle Environment (MPE) (Mordatch and Abbeel 2018). We conduct 5 random seeds for each algorithm and report the 1st, median, and 3rd quartile results. Due to space limitations on pages, we leave the MPE experiments in the Appendix.

Experiments Setup

StarCraft II Micromanagement The StarCraft Multi-Agent Challenge (Samvelyan et al. 2019) is a popular benchmark in cooperative multi-agent environments, where agents must form groups and work together to attack built-in AI enemies. The controlled units only access local observations within a limited field of view and take discrete actions. At each time step, every agent takes an action, and then the environment feedbacks a global team reward, which is computed by the accumulative damage point to the enemies. To evaluate the efficacy of different algorithms, we employ the training paradigm as previous notable works (Du et al. 2019; Zhou et al. 2020a) which utilizes 32 parallel runners to generate trajectories and store them into batches.

Google Research Football The Academy scenarios of the Google Research Football environment (Kurach et al. 2020) are inherently cooperative tasks that simulate partial football match scenes. We use the *Floats* (115-dimensional vector) observation setting including *players' coordinates, ball possession, ball direction, active player, and game mode.* The GRF is a highly sparse reward benchmark because it only feedbacks a global team reward r in the end, i.e., +1 bonus when scoring a goal and -1 bonus when conceding one.

Performance Comparisons

To demonstrate the effectiveness of IAM, we combine it with two representative CTDE algorithms: QMIX and VDN, which represent two ways of value factorization, i.e., linear

[1] Please note that although $\mathcal{L}_i^E$ in 6 is the same across different agents, their corresponding computed gradients are different, which is detailed in the Appendix.

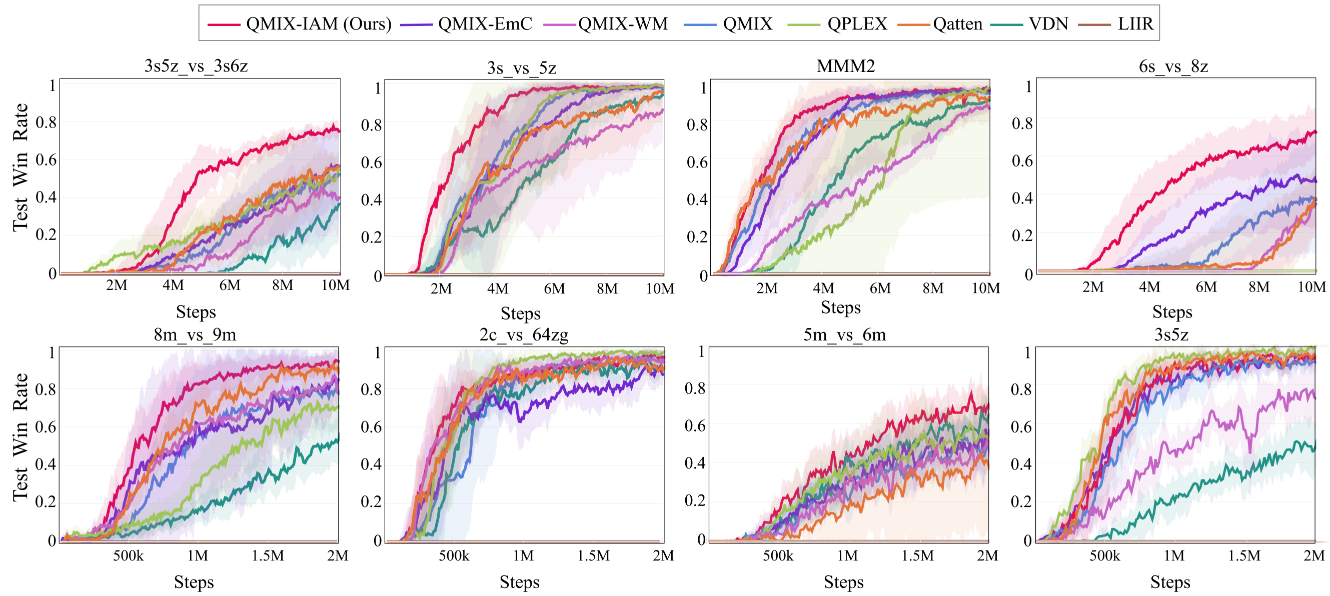


Figure 4: Performance comparisons for various maps in SMAC.

and non-linear. We denote them as QMIX-IAM and VDN-IAM respectively. To compare the performance of different rewards, we choose different types of reward-shaping methods as baselines: (1) Curiosity-based intrinsic rewards in EMC (Zheng et al. 2021). It's a representative CTDE's reward-shaping method based on curiosity. To fairly compare the impact of rewards, we remove its episodic memory and incorporate it with VDN and QMIX denoted by VDN-EmC and QMIX-EmC respectively. (2) Add world model based reward (Ma et al. 2022) into RA-CTDE, denoted by VDN-WM and QMIX-WM. (3) LIIR (Du et al. 2019) that utilizes learned intrinsic rewards. The remaining baselines are CTDE algorithms: (4) QPLEX. (5) Qatten. (6) QMIX. (7) VDN. For ease of comparison, we separate the performance comparison of QMIX-IAM and VDN-IAM, details on the latter are demonstrated in the Appendix.

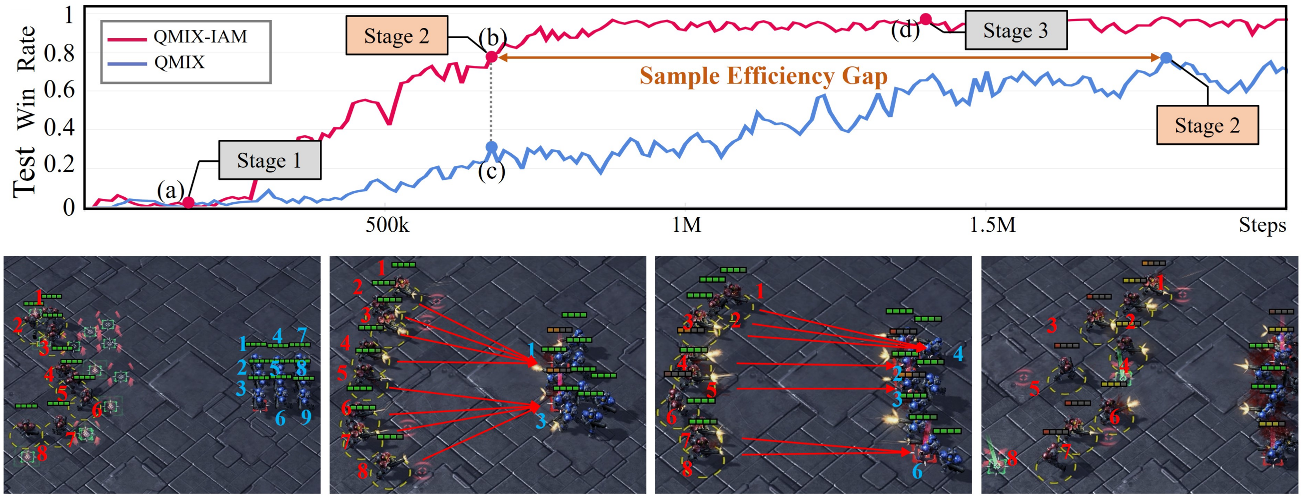
QMIX-IAM outperforms baselines. As shown in Figure 4, the performance of QMIX has been significantly improved after using the action model based reward, and QMIX-IAM outperforms other baselines in most scenarios, especially on several very hard maps requiring strong team cooperation. It indicates that the action model based reward can encourage consistent policy behaviors among agents and improve the performance of the CTDE algorithm. When using the exploration-based reward alone, QMIX-EmC only achieves performance improvements over QMIX on *6h_vs_8z* and *3s_vs_5z*, which indicates that the exploration-based reward lacks generalization for cooperative tasks. Based on the world model intrinsic reward, the performance of QMIX only has performance improvements on *3s5z*. This indicates that the world model based reward cannot generalize well in complex scenarios for high-dimensional observations and lacks in reflecting agents' action tendencies. Besides, QMIX-IAM also performs better than prominent CTDE methods, i.e. QPLEX and Qatten.

Strengths of IAM

An explicable example of IAM's impact. We visualize an illuminating map 8m_vs_9m in Figure 5, demonstrating how the IAM improves QMIX's performance by action tendency consistency. Among 3 training policy stages, QMIX takes the most samples to achieve Stage 2. The essential reason is that agents cannot reach team unanimity when attacking, causing their dispersion of firepower. Under the guidance of our action model intrinsic reward, agents will take the initiative to cultivate a tacit understanding of each other's action tendencies. Then agents can quickly achieve the team's consistent goal with only a few training samples, thus greatly improving sample efficiency than QMIX.

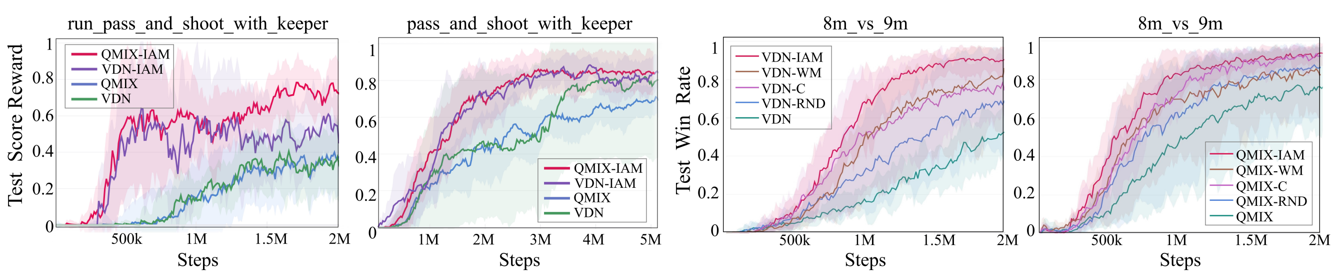
IAM can also obtain improved performance in highly sparse reward environments. To evaluate IAM's performance in deeply sparse reward environments, we choose two challenging tasks from GRF including *Academy_run_pass_and_shoot_with_keeper* and *Academy_pass_and_shoot_with_keeper*. We choose QMIX and VDN as baselines. As shown in Figure 6 (a) and (b), our method can significantly enhance the performance of QMIX and VDN, which indicates that IAM generalizes well in sparse-reward environmental scenarios.

Ablation: Our proposed intrinsic reward outperforms others when using RA-CTDE. In order to compare the performance of different rewards added in RA-CTDE, we compare IAM with additional baselines: (1) Add curiosity based reward in EMC, denoted by VDN-C and QMIX-C. (2) Add random network distillation(RND) reward into RA-CTDE, denotedy VDN-RND and QMIX-RND. We conduct these algorithms in *8m_vs_9m* and demonstrate results in



(a) Stage 1: Spread out (b) Stage 2: Concentrate firepower (c) Divergent firepower (d) Stage 3: Flexibly escape

Figure 5: A visualization example of IAM on 8m_vs_9m. In this task, agents need to obtain a three-stage team policy to win. In stage 1, agents need to be spread out to maximize the distraction of enemy attacks. In stage 2, agents need to maximize the concentration of firepower on the same enemy and reduce the enemy's numbers. In stage 3, agents need to escape quickly when they are low on blood to avoid being attacked and increase survival time. Among them, stage 2 is the hardest to learn because the agents need to cooperate to achieve the same policy target, i.e. action tendency consistency. (a), (b), and (d) represent three team policy stages of QMIX-IAM. (d) exhibits the distributed fire against the enemy of QMIX.



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

Figure 6: Ablation experiments. (a) and (b) show the performance comparison in two scenes of GRF. (c) and (d) exhibit the performance comparison in RA-CTDE combined with different rewards.

Figure 6. Both VDN-IAM and QMIX-IAM outperform others which implies that predictive information about action is beneficial for cooperation. Besides, after using RA-CTDE, all these intrinsic rewards have achieved performance improvements, indicating that the way RA-CTDE uses intrinsic rewards is reasonable and provides a new direction for CTDE to utilize intrinsic rewards. Besides, Exploration-based rewards don't perform as well as the action model based rewards, which indicates that the RA-CTDE framework can use different intrinsic rewards but the cooperative intrinsic rewards perform better than the exploration one in cooperative multi-agent systems.

Besides the aforementioned experiments, we also conduct ablation experiments to demonstrate the outperformance of RA-CTDE's reward-adding manner and RA-CTDE's equivalence to CTDE, which are detailed in the Appendix.

Conclusions and Limitations

We find that the CTDE algorithm suffers from low sample efficiency and attribute it to the team consensus inconsistency among agents. To tackle this problem, we design a novel intrinsic action model based reward and transform the CTDE into an equivalent variant, RA-CTDE. Then we use a novel integration of intrinsic rewards with RA-CTDE. Since our action model intrinsic rewards can boost consistent team policy and our proposed RA-CTDE can flexibly use calculated intrinsic rewards, our method shows significant outperformance on challenging tasks in the SMAC and GRF benchmarks. The limitations of our work are that we did not consider environments with continuous state-action space and did not make specific designs for heterogeneous agents. For future work, we will conduct additional research in the aforementioned directions.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (2022ZD0116402), NSFC 62273347, Jiangsu Key Research and Development Plan (BE2023016).

References

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Dębiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research*, 27(4): 819–840.

Böhmer, W.; Rashid, T.; and Whiteson, S. 2019. Exploration with unreliable intrinsic reward in multi-agent reinforcement learning. *arXiv preprint arXiv:1906.02138*.

Cao, Y.; Yu, W.; Ren, W.; and Chen, G. 2012. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1): 427–438.

Chitnis, R.; Tulsiani, S.; Gupta, S.; and Gupta, A. 2020. Intrinsic motivation for encouraging synergistic behavior. *arXiv preprint arXiv:2002.05189*.

Dahiya, A.; Aroyo, A. M.; Dautenhahn, K.; and Smith, S. L. 2023. A survey of multi-agent Human–Robot Interaction systems. *Robotics and Autonomous Systems*, 161: 104335.

Du, Y.; Han, L.; Fang, M.; Liu, J.; Dai, T.; and Tao, D. 2019. Liir: Learning individual intrinsic reward in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

Hernandez-Leal, P.; Kartal, B.; and Taylor, M. E. 2019. Agent modeling as auxiliary task for deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, volume 15, 31–37.

Hüttenrauch, M.; Šošić, A.; and Neumann, G. 2017. Guided deep reinforcement learning for swarm systems. *arXiv preprint arXiv:1709.06011*.

Iqbal, S.; and Sha, F. 2019. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. *arXiv preprint arXiv:1905.12127*.

Jaques, N.; Lazaridou, A.; Hughes, E.; Gulcehre, C.; Ortega, P.; Strouse, D.; Leibo, J. Z.; and De Freitas, N. 2019. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, 3040–3049. PMLR.

Kurach, K.; Raichuk, A.; Stańczyk, P.; Zając, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4501–4510.

Lin, F.; Huang, S.; Pearce, T.; Chen, W.; and Tu, W.-W. 2023. TiZero: Mastering Multi-Agent Football with Curriculum Learning and Self-Play. *arXiv preprint arXiv:2302.07515*.

Ma, Z.; Wang, R.; Li, F.-F.; Bernstein, M.; and Krishna, R. 2022. ELIGN: Expectation Alignment as a Multi-Agent Intrinsic Reward. *Advances in Neural Information Processing Systems*, 35: 8304–8317.

Matignon, L.; Laurent, G. J.; and Le Fort-Piat, N. 2012. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1): 1–31.

Mguni, D. H.; Jafferjee, T.; Wang, J.; Slumbers, O.; Perez-Nieves, N.; Tong, F.; Yang, L.; Zhu, J.; Yang, Y.; and Wang, J. 2021. Ligs: Learnable intrinsic-reward generation selection for multi-agent learning. *arXiv preprint arXiv:2112.02618*.

Mordatch, I.; and Abbeel, P. 2018. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Motlaghzadeh, K.; Eyni, A.; Behboudian, M.; Pourmoghim, P.; Ashrafi, S.; Kerachian, R.; and Hipel, K. W. 2023. A multi-agent decision-making framework for evaluating water and environmental resources management scenarios under climate change. *Science of The Total Environment*, 864: 161060.

Mushtaq, A.; Haq, I. U.; Sarwar, M. A.; Khan, A.; Khalil, W.; and Mughal, M. A. 2023. Multi-Agent Reinforcement Learning for Traffic Flow Management of Autonomous Vehicles. *Sensors*, 23(5): 2373.

Oliehoek, F. A.; and Amato, C. 2015. A concise introduction to decentralized pomdps.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1): 7234–7284.

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*.

Samvelyan, M.; Rashid, T.; De Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

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

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, 5887–5896. PMLR.

Strouse, D.; Kleiman-Weiner, M.; Tenenbaum, J.; Botvinick, M.; and Schwab, D. J. 2018. Learning to share and hide intentions using information regularization. *Advances in neural information processing systems*, 31.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo,

J. Z.; Tuyls, K.; et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.

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

Wang, J.; Ren, Z.; Han, B.; Ye, J.; and Zhang, C. 2020a. Towards understanding linear value decomposition in cooperative multi-agent q-learning.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2020b. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*.

Wang, T.; Wang, J.; Wu, Y.; and Zhang, C. 2019. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*.

Wiering, M. A.; et al. 2000. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, 1151–1158.

Zhang, S.; Cao, J.; Yuan, L.; Yu, Y.; and Zhan, D.-C. 2023. Self-Motivated Multi-Agent Exploration. *arXiv preprint arXiv:2301.02083*.

Zheng, L.; Chen, J.; Wang, J.; He, J.; Hu, Y.; Chen, Y.; Fan, C.; Gao, Y.; and Zhang, C. 2021. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34: 3757–3769.

Zhou, M.; Liu, Z.; Sui, P.; Li, Y.; and Chung, Y. Y. 2020a. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 11853–11864.

Zhou, M.; Luo, J.; Villella, J.; Yang, Y.; Rusu, D.; Miao, J.; Zhang, W.; Alban, M.; Fadakar, I.; Chen, Z.; et al. 2020b. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. *arXiv preprint arXiv:2010.09776*.