Efficient Credit Assignment in Cooperative Multi-Agent Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Cooperative multi-agent reinforcement learning (MARL) algorithms are crucial in addressing real-world challenges wherein multiple agents collaborate to achieve common objectives. The effectiveness of these algorithms hinges on the accurate estimation of agent action values, typically attained through learning joint and individual action values. However, challenges arise due to the credit assignment problem since it is difficult to accurately attribute the global reward to the actions of individual agents, which limits sample efficiency. This paper introduces ECA, an episodic control-based method, to mitigate this limitation by directly evaluating and assigning individual agent credits. ECA leverages episodic memory to store and cluster past interaction experiences between agents and the environment. Building upon these experiences, we introduce an intrinsic reward signal, quantifying the individual agent credits to the joint goal. This proposed reward signal serves as a corrective measure to revise individual action values, thereby improving the accuracy of individual and joint value estimations. We evaluate our methodology on StarCraft multi-agent challenge (SMAC) and Google Research Football (GRF) tasks, demonstrating that our method significantly improves the sample efficiency of state-of-the-art cooperative MARL algorithms.

1 Introduction

Cooperative MARL has emerged as a pivotal paradigm for addressing complex real-world challenges such as autonomous driving (Shalev-Shwartz et al., 2016), traffic control(Wiering et al., 2000), robotics(Gupta et al., 2017; Wang et al., 2020a), health care(Abdellatif et al., 2021) and finace(Zhang et al., 2021a), where multiple agents need to achieve shared objectives by collaborating with teammates. Most MARL algorithms follow a dual-phase strategy including centralized training and decentralized execution (CTDE). The training of agents is centralized, typically using a centralized critic; while execution or decision-making occurs decentralized during deployment(Oliehoek et al., 2008; Kraemer & Banerjee, 2016; Rashid et al., 2020). Based on the CTDE paradigm, MARL algorithms(Sunehag et al., 2017; Rashid et al., 2020; Wang et al., 2021a) achieved great success in solving cooperative multi-agent tasks.

For CTDE paradigm, accurate credit assignment (Nguyen et al., 2018; Zhang et al., 2021a; Zhou et al., 2020) is a crucial problem for effective policy optimization. It allows agents to identify and reinforce actions that lead to desirable team success while discouraging actions that focus on individual success and harm the team proceeds. In contrast, inaccurate credit assignment introduces various issues that harm sample-efficient policy optimization, such as a lack of coordination among multiple agents (Sukhbaatar et al., 2016), suboptimal joint policy(Zhou et al., 2023), and increased exploration challenges (Zheng et al., 2021). Therefore, it is important to explore the problem of accurate credit assignment.

However, ground truths for credit assignments are typically unavailable, and even domain experts find defining the credits both challenging and time-consuming. Several methods have been proposed to address these challenges in credit estimation. They can be divided into two categories, implicit credit assignment and explicit credit assignment. Implicit credit assignment focuses on learning associations between agents and observed rewards without explicitly linking rewards to individual actions (Lowe et al., 2017; Zhou et al., 2020; Wang et al., 2021b; Zhang et al., 2021b; Peng et al., 2021; Su et al., 2021). These methods typically employ expressive value functions to coordinate agents. Unfortunately, existing studies (Rashid et al., 2020; Wang et al., 2021a; Chen et al., 2023) revealed that implicit credit assignment has the limitation of monotonic that can induce inaccurate individual estimation. On the other side, explicit credit assignment directly computes and attributes credits to specific actions taken by each agent (Foerster et al., 2018; Wang et al., 2020b; Proper & Tumer, 2012). However, the accuracy of credit assignments remains limited due to the complex environments, high-dimensional state spaces, and scarce background information.

To tackle the above limitations, in this paper, we introduce a simple yet effective episodic control method, named **ECA**, to conduct **E**fficient **C**redit **A**ssignment for more sample-efficient MARL policy optimization. Inspired by the paradigm of episodic control (Lengyel & Dayan, 2007; Pritzel et al., 2017; Blundell et al., 2016), the spirit of ECA is to utilize the *past interaction experiences* between agents and environments to help estimate the agent credits under certain states. However, different from existing episodic control works (Zheng et al., 2021; Na et al., 2024) that focus on measuring the global credits, ECA supports more fine-grained and agent-level explicit credit assignment. Overall, ECA contains two phases, episodic memory construction and episodic control. In the first phase, we reduce the dimension of concrete global states and discretize them into unique and countable abstract states, thus concrete states can be naturally grouped based on abstract topological similarity. After that, leveraging statistics of past trajectories with the same abstract states, we calculate the average expectation of episodic returns for each agent, serving as individual credits. In the second episodic control phase, we integrate these credits to correct individual values during policy optimization.

The essence of ECA lies in the direct measurement and correction of the individual agent's impact on the team, thereby improving the sample efficiency of MARL policy optimization. Experimental evaluations have been conducted on SMAC (Samvelyan et al., 2019) and GRF (Kurach et al., 2019) tasks involving comparisons with existing MARL algorithms and episodic control-based methods. The results convincingly demonstrate that credit assignment significantly improves sample efficiency. The ECA code repository is publicly available for further reproduction[1]. In summary, this work makes the following contributions:

- The paper introduces ECA, a new episodic control method designed to improve sample efficiency in cooperative MARL by enabling more accurate credit assignment to individual agents.
- ECA provides fine-grained, explicit credit assignments at the agent level, allowing for more precise policy optimization.
- The paper validates ECA through experiments on standard MARL benchmarks, namely SMAC and GRF. Results show that ECA outperforms existing MARL algorithms and episodic control-based methods, demonstrating significant improvements in sample efficiency.
- We make the ECA code repository publicly available, promoting transparency and enabling other researchers to reproduce and build upon their work.

2 Preliminaries

2.1 Multi-agent Reinforcement Learning

MARL builds upon the Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs) (Oliehoek & Amato, 2016) framework where multiple agents interact within an environment characterized by partial observability. A Dec-POMDP is a tuple $\langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, n, \gamma \rangle$, where $\mathcal{I}$ is a set of n agents $(I_0, ..., I_i, ..., I_{n-1})$ which take the observations $\mathbf{o}^t$ as inputs. $\mathcal{S}$ represents the state space constructed by states across different time steps s^t, where t is the time step. $\mathcal{A}$ denotes the action space of the agents thus the joint action $\mathbf{a} \in \mathcal{A}^n$. $\mathcal{P}(s^{t+1}|s^t, \mathbf{a}^t)$ denotes the transition function of state s^t and joint action $\mathbf{a}^t$, $\mathcal{R}(s^{t+1}|s^t, \mathbf{a}^t)$ is the reward function. $\mathbf{o}^t = \mathcal{O}(s^t, I)$ is the observations of all agents, and γ is the discount factor. Dec-POMDPs encapsulate the inherent complexity of decentralized decision-making, where agents

[1] `https://anonymous.4open.science/r/ECA-Repo`

make informed choices based on their local observations while considering the actions and observations of their peers. And the interaction trajectory can be represented as:

$$(s^0, \mathbf{o}^0, \mathbf{a}^0, r^0), ..., (s^t, \mathbf{o}^t, \mathbf{a}^t, r^t), ..., (s^T, \mathbf{o}^T, \mathbf{a}^T, r^T) \tag{1}$$

Centralized Training and Decentralized Execution (CTDE) (Bernstein et al., 2002; Oliehoek et al., 2008; Gupta et al., 2017) is a prominent paradigm in MARL that addresses the challenges of coordinating multiple agents while optimizing their policies. The key characteristic of CTDE lies in its two-phase strategy, where training is performed centrally while execution or decision-making occurs in a decentralized manner during deployment. During the training process, at time-step t, the whole team cooperates to find the optimal joint action-value function $Q^*_{tot}(s, \mathbf{a})$:

$$Q^*_{tot}(s^t, \mathbf{a}^t) = r(s^t, \mathbf{a}^t) + \gamma \mathbb{E}\left[\max Q^*_{tot}(s^{t+1}, \mathbf{a}^{t+1})\right] \tag{2}$$

During execution, the learned policies are employed independently, allowing each agent to make decentralized decisions based on local observations. CTDE balances centralized knowledge aggregation and decentralized decision-making, providing a practical framework for addressing complex coordination challenges in MARL scenarios. However, the CTDE framework typically relies on global rewards obtained by the team for value estimation and policy optimization, which could not provide accurate agent-level credit assignments.

2.2 State Abstraction

In a high-dimensional state space that contains infinite concrete states, each concrete state s can be represented as $(\phi^0, ..., \phi^d, ...)$. Each component ϕ^d is a scalar value and has lower and upper boundaries $\phi^d \in [l_d, u_d]$, which is determined by the environment. We split each dimension into N equal intervals, and the concrete value ϕ^d falls into an interval η_d:

$$\eta_d = \lfloor \frac{u_d - l_d}{N} * (\phi^d - l_d) \rfloor \tag{3}$$

In this way, a high-dimensional concrete state s can be represented as an abstract state $\hat{s} = (\eta_0, \dots, \eta_d, \dots)$, which encodes each dimension of the concrete state to an integer and is easy to count. Similar concrete states are grouped into the same abstract state if their concrete values fall within the same intervals. This approach allows for measuring the effects of different actions by agents within the same abstract state, enabling quantifying their contributions at the abstract state level.

2.3 Reward Confidence Score

The reward confidence score efficiently measures the episodic returns resulting from the states. The computation of the reward confidence score is based on the statistic of episodic returns. Given the trajectory ψ_k:

$$\psi_k = \{(\hat{s}^0, \mathbf{o}^0, \mathbf{a}^0, r^0), ..., (\hat{s}^t, \mathbf{o}^t, \mathbf{a}^t, r^t), ..., (\hat{s}^T, \mathbf{o}^T, \mathbf{a}^T, r^T)\} \tag{4}$$

T is the length of the trajectory ψ_k, note $\hat{s}^t$ is an abstract state at time step t, the episodic return $\mathcal{E}_k$ of the trajectory ψ_k is computed as follows:

$$\mathcal{E}_k = \sum_{t=0}^{T} r^t \tag{5}$$

Thus the reward confidence score $rcs^t(\hat{s}^t)$ of global state $\hat{s}^t$ can be computed by the following method:

$$rcs^t(\hat{s}^t) = \frac{1}{|\Psi|} \sum_{k=0}^{|\Psi|} \mathcal{E}_k, \hat{s}^t \in \Psi \tag{6}$$

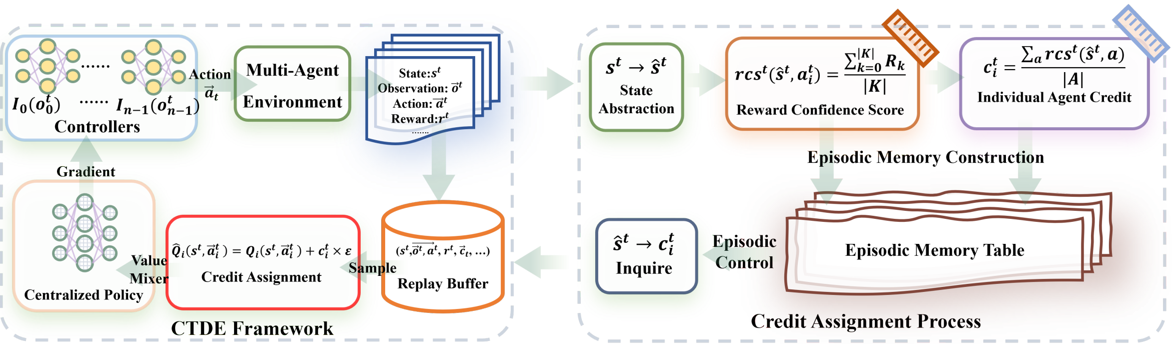
$I_0(o_0^t)$ …… $I_{n-1}(o_{n-1}^t)$
Controllers
Action $\vec{a}_t$
Multi-Agent Environment
State: s^t
Observation: $\vec{o}^t$
Action: $\vec{a}^t$
Reward: r^t
……
Gradient
Centralized Policy
Value Mixer
$\hat{Q}_i(s^t, \vec{a}_i^t) = Q_i(s^t, \vec{a}_i^t) + c_i^t \times \varepsilon$
Credit Assignment
Sample
$(s^t, \overrightarrow{o^t, a^t}, r^t, \vec{c}_t, \ldots)$
Replay Buffer
CTDE Framework
$s^t \rightarrow \hat{s}^t$
State Abstraction
$rcs^t(\hat{s}^t, a_i^t) = \frac{\sum_{k=0}^{|K|} R_k}{|K|}$
Reward Confidence Score
$c_i^t = \frac{\sum_a rcs^t(\hat{s}^t, a)}{|A|}$
Individual Agent Credit
Episodic Memory Construction
Episodic Memory Table
Episodic Control
$\hat{s}^t \rightarrow c_i^t$
Inquire
Credit Assignment Process

Figure 1: An overview of ECA workflow.

where Ψ is the collection of trajectories that encountered the state $\hat{s}^t$. The reward confidence score $rcs^t(\hat{s}^t)$ represents a *quality* measurement of global state $\hat{s}^t$. Specifically, a higher $rcs^t(\hat{s}^t)$ indicates that the team earns more returns when encountering $\hat{s}^t$.

In previous reward confidence score-based policy optimization (Li et al., 2023), $rcs^t(\hat{s}^t)$ is used as an explicit signal and added to the global reward for enhancing the joint action value Q^*_{tot}:

$$\begin{aligned} Q^*_{tot}(s^t, \mathbf{a}^t) = {} & r(s^t, \mathbf{a}^t) \\ & + \gamma \mathbb{E}\left[\max Q^*_{tot}(s^{t+1}, \mathbf{a}^{t+1})\right] \\ & + rcs^t(\hat{s}^t) \end{aligned} \tag{7}$$

3 Methodology

This section introduces the details of the ECA methodology. The core concept of ECA is to estimate agent contributions based on past experiences in various states. Specifically, we calculate the episodic returns achieved for actions taken by each agent in particular states. ECA comprises two main steps: episodic memory construction and episodic control. Figure 1 illustrates the ECA workflow, built upon the classic CTDE framework.

In the episodic memory construction phase, ECA begins by discretizing the concrete states into unique and countable abstract states (*i.e.*, State Abstraction, as introduced in Section 2.2). Specifically, the states are grouped and clustered by their abstract topological similarity. Following this state abstraction, agent credits for each abstract state are calculated (i.e., C_i^t) and stored in an episodic memory table. In the episodic control phase, these stored agent credits are retrieved to efficiently adjust individual action values and enhance centralized policy optimization (Schaul et al., 2015).

3.1 Episodic Memory Construction

To measure agent credit, we quantify the contribution of each agent's actions in specific states based on past experiences. However, the concrete states are high-dimensional and continuous, leading to an infinite number of states, which poses challenges for state-based credit analysis. To mitigate this, we adopt a topological state clustering method for state abstraction, which discretizes the states and merges similar concrete states together. In this way, we divide infinite state space into finite groups and perform the credit analysis of agent actions on each group.

State-based Credit Measurement. The past experiences, represented by K trajectories (*i.e.*, all the concrete states, actions, and rewards in a trajectory), provide valuable information for assessing the contribution of each agent's actions. In MARL, each trajectory can be represented as $\psi = \{(\hat{s}^0, \mathbf{o}^0, \mathbf{a}^0, r^0), ..., (\hat{s}^t, \mathbf{o}^t, \mathbf{a}^t, r^t), ...\}$, where $\hat{s}^t$ is an abstract state (based on interval abstraction) of concrete state s^t shared by all the agents at time step t, $\mathbf{o}^t$ contains the respective observation of each agent,

Algorithm 1 Episodic Memory Construction

```
Input: $\mathcal{I}$: decentralized agents, $n$: number of agents, $E$: Dec-POMDP environment, $\mathcal{M}$: episodic memory, $Steps$: number of time steps.
Output: $\mathcal{B}$: Replay Buffer with episodic data and our individual credits.
 1: Initialize: $\mathcal{B}, \mathcal{F} \leftarrow \emptyset, t = 0$
 2: while $t < Steps$ do
 3:     $s^t, \mathbf{o}^t \leftarrow E$.Reset()
 4:     $\hat{s} \leftarrow \mathcal{M}$.State_Abstraction($s^t$)        ▷ Convert concrete states to unique labels.
 5:     $\mathbf{a}^t \leftarrow \mathcal{I}$.Agents_Action_Selection($\mathbf{o}^t$)        ▷ Decentralized agents select actions for execution.
 6:     $s^{t+1}, \mathbf{o}^{t+1}, r^t \leftarrow E$.Step($\mathbf{a}^t$)
 7:     $\langle c_0, ..., c_i, ..., c_{n-1} \rangle \leftarrow \mathcal{M}$.Update($\hat{s}, \mathbf{a}^t, r^t$)        ▷ Individual estimation of all the agents.
 8:     $\mathcal{B}$.append($s^t, \mathbf{o}^t, \mathbf{a}^t, r^t, \langle c_0, ..., c_i, ..., c_{n-1} \rangle$)        ▷ Store the individual credits into replay buffer $\mathcal{B}$.
 9:     $t \leftarrow t + 1$
10: end while
```

and $\mathbf{a}^t$ is the joint action of the team at time step t. For each trajectory ψ, the episodic return $\mathcal{E}_\psi$ is calculated by the sum of the rewards across all the time steps, details can be found in Section 2.3. For each action a_i (taken by the i^{th} agent) under a specific state $\hat{s}$, we measure its *contribution* by counting the average of the episodic return of the trajectories that cover the state $\hat{s}$. The contribution is represented by the reward confidence score:

$$rcs(\hat{s}, a_i) = Avg(\{\mathcal{E}_\psi | \psi \in \Psi \wedge cover((\hat{s}, a_i), \psi)\}) \tag{8}$$

where Ψ represents all the trajectories, and *cover* indicates that the trajectory ψ includes the state $\hat{s}$ and the action a_i^t taken by the i^{th} agent in that state. Intuitively, the contribution of the action a_i^t under the state $\hat{s}$ is determined by the average episodic return of the trajectories that include this action in that state.

As an agent may take different actions in the same abstract state across various trajectories, ECA calculates the reward confidence scores for all actions within the state to estimate each agent's impact on state $\hat{s}$. We assess the individual credit of the i^{th} agent under a specific state $\hat{s}$:

$$c_i(\hat{s}) = Avg(\{rcs(\hat{s}, a_i) | a_i \in \mathcal{A}\}) \tag{9}$$

where $\mathcal{A}$ represents all possible actions of the i^{th} agent in state $\hat{s}$. With the individual credit c_i, we can measure the influence of the agents on the episodic returns. A higher individual credit reflects a stronger impact of the individual decision-making on the team.

Algorithm 1 summarizes the processes of Episodic Memory Construction. We utilize episodic data to measure individual credits first, then store them in a tabular episodic memory. Specifically, we transform each concrete state s^t into an abstract state $\hat{s}$ via topological state abstraction (line 4). Following the action selection for each agent and decentralized execution (line 7), we input all abstract states, actions, and rewards into the episodic memory module to compute the individual credit of each agent at every time step (*i.e.*, under the state at the specific time step). Subsequently, individual credits are deposited into the replay buffer $\mathcal{B}$ for subsequent policy optimization (line 8).

3.2 The episodic memory structure of ECA

Figure 2 illustrates the structure of episodic memory within ECA. Episodic memory in ECA is structured as a nested key-value table. Specifically, it utilizes an abstract state as the primary key, facilitating precise retrieval and efficient memory operations. Nested within are agent IDs, with each agent having a corresponding credit value. Furthermore, the secondary nested keys represent the actions undertaken by each agent. These actions are associated with three values: the occurrence of abstract state and agent action pairs $(\hat{s}, a_i^t)$, the cumulative episodic returns within the trajectory $\mathcal{E}\psi$ containing $(\hat{s}, a_i^t)$, and a reward confidence score denoting the average episodic returns, detailed in Equation 8.

During memory construction (as outlined in Algorithm 1), interaction experiences are gathered at each time step t. Initially, the concrete state s_t is transformed into an abstracted state $\hat{s}$ (line 4). Subsequently, occurrences of each abstract state and agent action pair $(\hat{s}, a_i^t)$ are recorded, alongside updates to corresponding

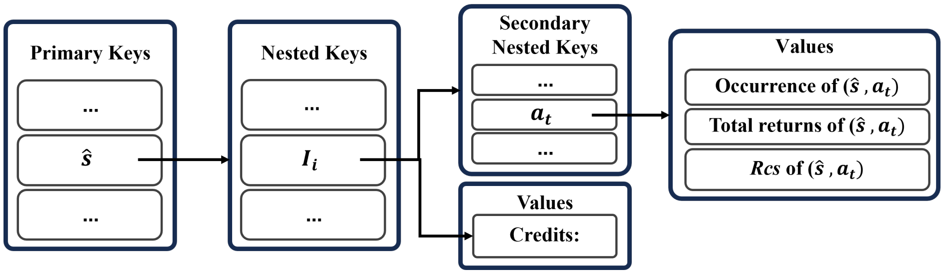

Figure 2: The specific structure of episodic memory in ECA. Our memory is a nested key-value table that records the agent credits and the information for credit computation.

Algorithm 2 Episodic Control-Based Credit Assignment

```
Input: $\mathcal{I}$: decentralized agents, $n$: number of agents, $\mathcal{P}$: centralized policy, $\mathcal{B}$: replay buffer, $\mathcal{Z}$: batch size, $\gamma$: discount factor, $\epsilon$: scaling factor in ECA.
Output: $Loss$: Optimal Centralized Policy.
 1: Initialize: $Loss = 0$
 2: while $u < \mathcal{Z}$ do
 3:     $s^u, \mathbf{o}^u, \mathbf{a}^u, r^u, \langle c_0^u, ..., c_i^u, ..., c_{n-1}^u \rangle \leftarrow \mathcal{B}$.Sample()
 4:     for $i = 0 \to n - 1$ do
 5:         $Q_i^u(o_i^u, a_i^u) \leftarrow \mathcal{I}_i$.Individual_Estimation$(o_i^u, a_i^u)$
 6:         $\hat{Q}_i^u(o_i^u, a_i^u) \leftarrow Q_i^u(o_i^u, a_i^u) + c_i \times \epsilon$        ▷ Revision of individual values by credits.
 7:     end for
 8:     $\hat{Q}_{tot}(s^u, \mathbf{a}^u) \leftarrow \mathcal{P}$.Value_Integration$(\hat{Q}_0^u, ..., \hat{Q}_{n-1}^u)$        ▷ Value integration operation, e.g., QMIX.
 9:     $\delta^u \leftarrow r^u + \gamma \max_{\mathbf{a}'} Q_{tot}(s', \mathbf{a}') - \hat{Q}_{tot}(s^u, \mathbf{a}^u)$        ▷ $s', \mathbf{a}'$ are the next state and action of $s^u, \mathbf{a}^u$.
10:     $Loss = Loss + \frac{1}{2}(\delta^u)^2$
11:     $u \leftarrow u + 1$
12: end while
```

cumulative episodic returns and reward confidence scores. With this information stored in episodic memory, agent credits are recalculated based on their actions' aggregate reward confidence scores, following Equation 9. This process ensures accurate approximation and retrieval of agent credits, crucial for further policy optimization (as outlined in Algorithm 2), enabling efficient episodic control and credit assignment within ECA.

3.3 Episodic Control

After measuring credit based on past trajectories, we apply the estimated credit in episodic control. The individual credits are used as intrinsic rewards for each agent through reward shaping (Ng et al., 1999; Burda et al., 2019; Zheng et al., 2021; Na et al., 2024):

$$\hat{Q}_i(o_i^t, a_i^t) = Q_i(o_i^t, a_i^t) + c_i \times \epsilon \tag{10}$$

where $Q_i(o_i^t, a_i^t)$ represents the individual action value of the i^{th} agent at time step t, and ϵ is a scalar to control the value scale of shaping reward. Note that the o_i^t can either be the current or historical observations. Using current observations allows agents to react promptly to the environment's current state and the actions of other agents, while historical observations provide valuable context and enable agents to learn from past experiences. In ECA, individual agents estimate the action values using the respective historical observations. After the reshaping of individual credits, the new individual value $\hat{Q}_i$ will be fed into a value integration module, such as mixture network in QMIX (Rashid et al., 2020), and then the value of $Q_{tot}^*(s^{t+1}, \mathbf{a}^{t+1})$ and Q_{tot}^* will be updated. Finally, we can obtain a revised joint action value for loss evaluation for centralized policy network update.

Algorithm 2 presents the details of Episodic Control. We enhance policy optimization by incorporating individual credits into original agent values (line 6). These adjusted individual agent values are then amal-

Table 1: The hyperparameters and policy network architecture of ECA.

Items	Values
Replay Buffer Size	5000
Batch Size	32
Discount Factor	0.99
Individual Agent Optimizer	Adam
Individual Agent Learning Rate	5e-4
Centralized Policy Optimizer	Adam
Centralized Policy Learning Rate	5e-4
Hidden Layer Size	64

gamated into joint-action values (line 8). Consequently, the centralized policy is trained based on the disparity between estimated joint-action values and the TD target, with the estimated joint-action values being adjusted using our individual credits.

4 Evaluation

4.1 Experiment setup

Dataset. We evaluate ECA on commonly used tasks SMAC (Samvelyan et al., 2019) and GRF (Kurach et al., 2019). SMAC and GRF provide various challenging multi-agent tasks, in which the agents need to experience millions of high-dimensional concrete states to learn a good policy. Specifically, we select six SMAC maps (Samvelyan et al., 2019) that varied by task difficulties, including an *Easy* task *1c3s5z*, two *Hard* tasks *3s_vs_5z* and *5m_vs_6m*, and three *Super Hard* tasks *3s5z_vs_3s6z*, *5s10z*, and *MMM2*, in where diverse agents (*e.g.*, *s* denotes *Stalkers* in the game) engage in team battles against common enemies. For the GRF task, we select academy_3_vs_1_with_keeper (*3_vs_1_WK*), academy_counterattack_easy (*CA_Easy*), and academy_counterattack_hard (*CA_Hard*), in which different numbers of players are controlled by the centralized policy to score in a football game under different conditions (Kurach et al., 2019).

Baseline. With the above tasks, we compare ECA with Curiosity-Driven Exploration (EMC) (Zheng et al., 2021) and Efficient episodic Memory Utilization (EMU) (Na et al., 2024), representative episodic control-based methods designed to improve the sample efficiency of existing cooperative MARL algorithms. EMC stores states in episodic memory, incorporates a curiosity module to measure the freshness of states, and leverages the measurement as a guide for exploration. Conversely, EMU focuses primarily on desired (highly-rewarded) state transitions, employing an episodic memory to store past experiences and measure corresponding expected returns as intrinsic rewards. We select Qplex (Wang et al., 2021a) as the base algorithm for EMC, EMU, and ECA to enable an adequate comparison. Moreover, the default value integration method is Qatten (Yang et al., 2020). The program of EMC and EMU is from the official and open code repositories, and the implementation of ECA is based on the PyMARL benchmark(Samvelyan et al., 2019).

Configurations. ECA is highly supplementary and straightforward, requiring no additional adaption on various MARL algorithms, rendering promising applicability and potential for solving more multi-agent tasks. For a fair comparison, we follow previous works (Zheng et al., 2021; Na et al., 2024) to set hyperparameters and network architecture, as listed in Table 1. Besides, ECA performs dimension reduction of concrete global states by multiplying a Gaussian random projection matrix, which makes smaller concrete global states and efficient state abstraction. These processed state vectors have a $(1, 24)$ shape on all the tasks. Subsequently, we set N in Equation 3 as 5. The scaling factor ϵ for shaping reward values in Equation 10 to 0.001 aligns with the values utilized in prior methods (Na et al., 2024). The other parameter choices also mirror the settings in previous works (Na et al., 2024), including the policy network architecture, learning rate, size of replay buffer, etc. We repeat each experiment five times to counteract randomness.

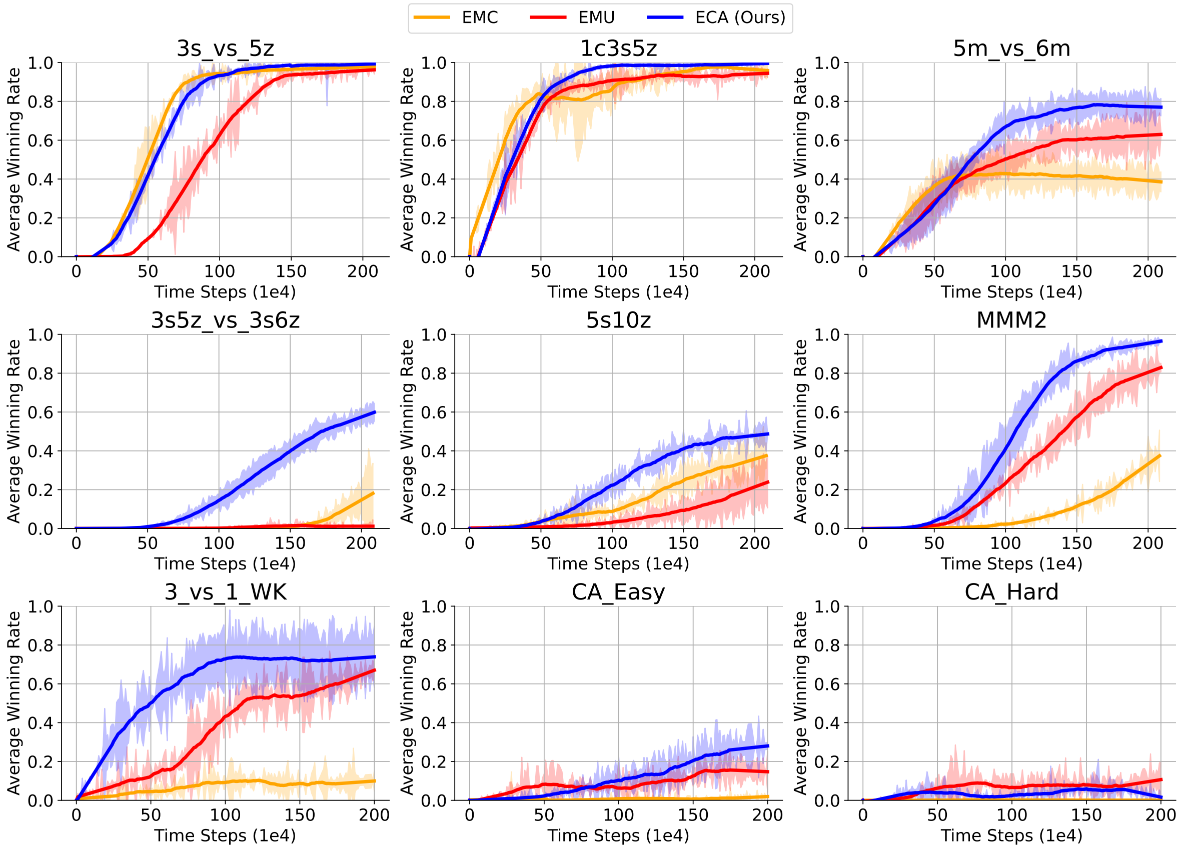

Figure 3: The evaluation of ECA and the state-of-the-art MARL episodic control methods.

4.2 Evaluation of ECA on large-scale multi-agent tasks

Figure 3 illustrates the trends in team winning rates during policy optimization by EMC, EMU, and our ECA. Table 2 lists the average winning rates achieved by policies trained after 2 million time steps. In SMAC tasks, while the winning rates of the three methods on the *Easy* task *1c3s5z* and the *Hard* task *3s_vs_5z* are close, ECA significantly outperforms EMC and EMU on *5m_vs_6m*, *3s5z_vs_3s6z*, *5s10z*, and *MMM2*, which are more challenging due to the number of agents and the strength of the enemies. In GRF tasks, ECA achieves 0.736 and 0.276 winning rates and outperforms other methods on *3_vs_1_WK* and *CA_Easy*, respectively. All methods, including EMC, EMU, and ECA, perform worse on the *CA_Hard* task in GRF due to the task difficulties. In summary, ECA enables a more sample-efficient MARL policy optimization on most designated tasks compared to the baselines.

An accurate individual value estimation determines the effectiveness of ECA. While obtaining ground-truth state values in complex environments is infeasible, our results show that accurate agent credit estimation offers effective guidance for determining agent contributions. The agent credits computed based on past experiences can be efficiently used to correct individual values in similar states, establishing effective knowledge transfer and minimizing the gap between current and optimal policies.

ECA produces an episodic memory table at each run involving 200,000 to 400,000 abstract states and the corresponding agent credits. The construction of episodic memory takes only a few minutes in total, which is negligible compared to the cost of MARL policy optimization. Based on the experimental performance and episodic memory size, ECA effectively and efficiently handles high-dimensional MARL environments.

Table 2: The average winning rates and standard deviations by EMC, EMU, and our ECA. Note that bold marks denote the highest winning rate achieved in the corresponding tasks.

Domains	Tasks	EMC	EMU	ECA (Ours)
SMAC	1c3s5z	0.961±0.011	0.943±0.011	**0.994±0.001**
	3s_vs_5z	0.977±0.005	0.964±0.003	**0.991±0.001**
	5m_vs_6m	0.388±0.034	0.627±0.044	**0.770±0.052**
	3s5z_vs_3s6z	0.162±0.066	0.012±0.006	**0.586±0.022**
	5s10z	0.365±0.090	0.226±0.110	**0.483±0.049**
	MMM2	0.350±0.065	0.816±0.050	**0.959±0.012**
GRF	3_vs_1_WK	0.097±0.028	0.657±0.031	**0.736±0.127**
	CA_Easy	0.017±0.007	0.148±0.045	**0.276±0.051**
	CA_Hard	0.009±0.006	**0.101±0.053**	0.023±0.010

4.3 Evaluation of ECA on different credit computation methods

The key component of ECA is to accurately evaluate the agent credits. Multiple alternatives to agent credit calculation can be considered. We compare our proposed method with other credit calculation methods that span different levels of detail when evaluating the impact of the agent's performance on the team. Reward confidence scores of global states (*i.e.*, *state_rcs*) measure the expected episodic returns of team cooperation without highlighting individual agent or action impacts (see Equation 6 in Section 2.3). Reward confidence scores of global state and local action pairs as defined in Equation 8 (*i.e.*, *state_action_rcs*) quantify the influence of individual actions on team episodic returns as defined in Equation 5. In contrast, *agent_credit* (our proposed method) outlined in Equation 9 measures an agent's contribution by considering all feasible actions in a state rather than focusing on individual actions.

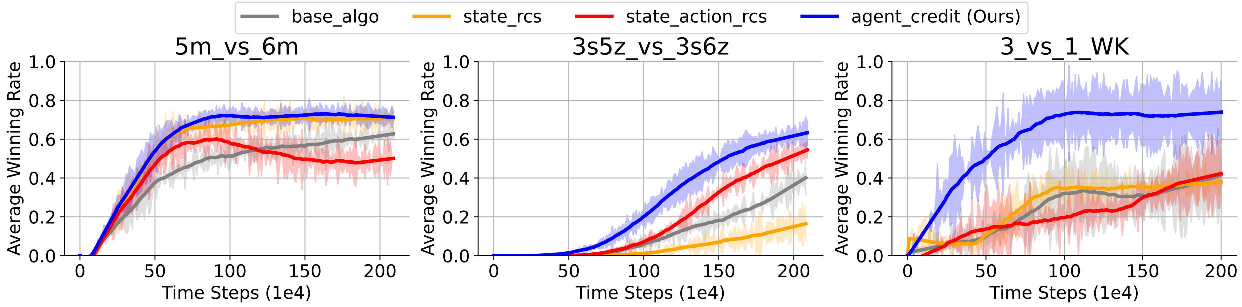

Figure 4: The winning rates of learned policies by incorporating different intrinsic rewards with the individual value estimation. Note that in *base_algo*, we do not perform credit assignment.

Figure 4 depicts the evaluation results of using different agent credits as intrinsic rewards. The results demonstrate that using agent credits as intrinsic rewards to revise the individual estimation significantly outperforms employing global intrinsic rewards, particularly on challenging tasks. Moreover, our method achieves higher winning rates than directly utilizing individual action reward confidence scores as intrinsic rewards. This indicates that our credits provide a more comprehensive and accurate value estimation for agents, whereas using action reward confidence scores might introduce variances due to their sparsity. In conclusion, directly assigning credits to agents and calculating individual credits is a better way for ECA.

4.4 Evaluation of ECA on various value integration methods

Another strength of ECA is that it can be adapted to different value integration methods. To showcase the adaptability of ECA, we apply it to different base algorithms with diverse value integration methods, including VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), and Qatten (Yang et al., 2020). We also evaluate EMC and EMU on the above value integration methods. Note that for the base algorithms of VDN, QMIX, and Qatten, we do not perform specific explicit reward shaping and credit assignment.

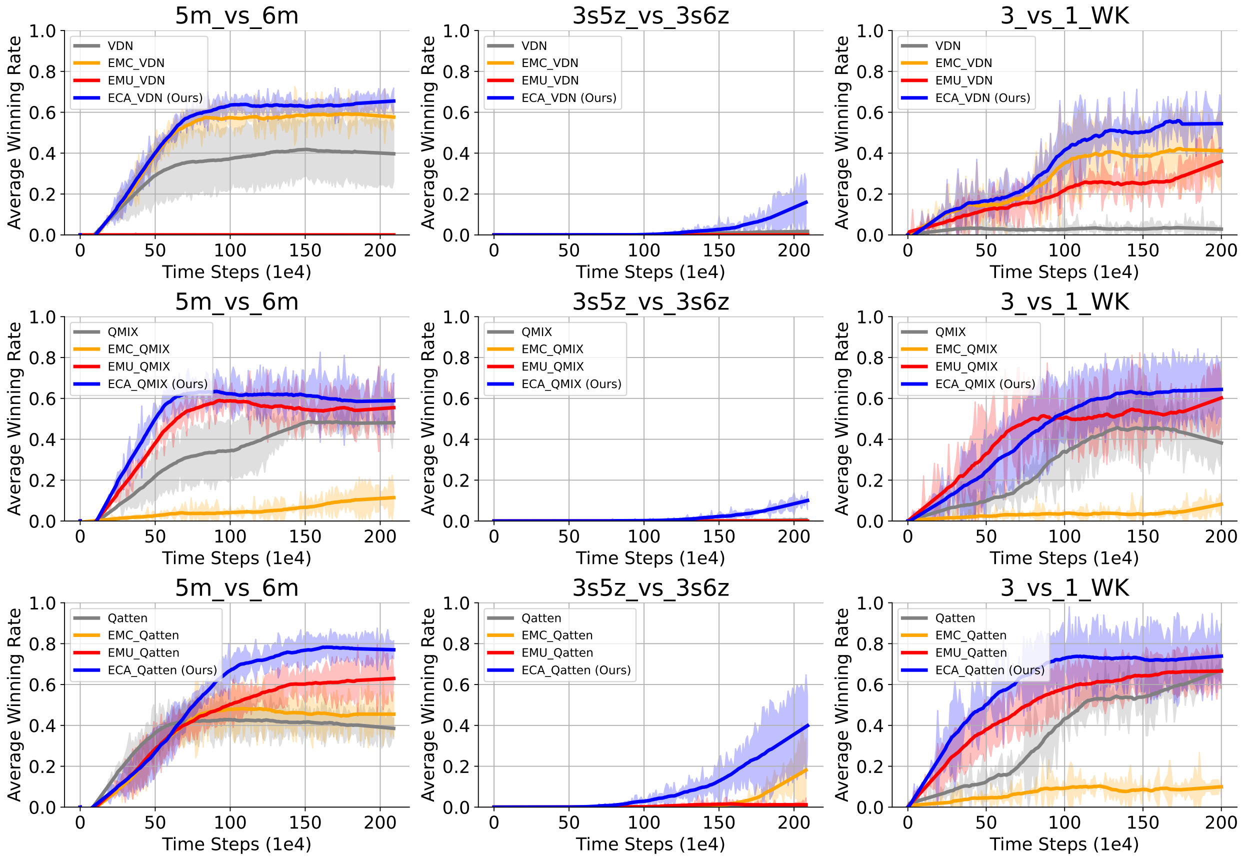

Figure 5: The winning rates of ECA across different value integration methods.

The results depicted in Figure 5 highlight a significant enhancement in sample efficiency when ECA is incorporated with these base algorithms. Such results indicate that ECA can be effortlessly integrated into existing MARL pipelines without requiring extensive modifications. Therefore, ECA is a versatile and practical method for sample efficient MARL. Moreover, the observed improvements in winning rates across various base algorithms also validate the accuracy of individual contribution credit measurements facilitated by ECA.

Moreover, we found that although EMC and EMU show significant performance improvement in most experiments with different integration methods, ECA outperforms them on all the tasks and establishes the most performance improvement against the base algorithms. Such results further prove the effectiveness of ECA in applying different multi-agent tasks and the potential of combining them with various base algorithms.

5 Related Work

5.1 Credit Assignment

Credit assignment in multi-agent settings involves attributing a global and shared reward from the environment to the individual actions of agents, ensuring that each agent's credit accurately reflects their contribution to coordinated performance. Solving inaccurate credit assignment problems has two directions, implicit and explicit credit assignment.

Implicit credit assignment aims to learn associations between agents and the observed rewards without explicitly attributing rewards to specific actions. Existing works have proposed novel implicit credit assignment structures. MADDPG (Lowe et al., 2017) learns individual policies by ascending value gradients. Value Decomposition Networks (VDN) (Sunehag et al., 2017) aggregates individual estimations to form the joint action value, while QMIX (Rashid et al., 2020) utilizes a mixer network to coordinate and combine individual values. Qatten (Yang et al., 2020) incorporates an attention mechanism to capture latent coordination relationships among agents, and Qplex (Wang et al., 2021a) employs a dueling network to circumvent direct optimization from monotonic assumptions. DOP (Wang et al., 2021b) decomposes the central critic into weighted individuals. FACMAC (Peng et al., 2021) employs a centralized gradient estimator for credit assignment. However, implicit credit assignment may be limited by constraints such as monotonicity (Rashid et al., 2020). Unlike implicit methods, explicit methods directly assign credits to agents. Difference rewards (DR) have been employed as an explicit method to enable agents to learn from a shaped reward by comparing the global reward to the reward obtained when an agent's action is replaced with a default action(Wolpert & Tumer, 2001; Tumer & Agogino, 2007; Proper & Tumer, 2012; Foerster et al., 2018; Tumer et al., 2002). However, DR could be more efficient due to the necessity for separate estimations of baselines and its diminished effectiveness in the presence of complex cooperative behaviors. Differently, COMA(Foerster et al., 2018) adopts agent-specific advantage functions to compute the agent credits, and Shapley Q-value is used to evaluate the possible expectation scale of available actions. Jiang *et al.* factorize and reshape team rewards into agent-specific rewards, thus accurately approximating heterogeneous agent contributions and improving energy efficiency(Jiang et al., 2023). However, learning accurate agent credit is still challenging due to the complex environments and agent interactions. Different from existing work, ECA focuses on agent-level explicit credit assignment and utilizes past interaction experiences to estimate the current credits of agents.

5.2 Neural Episodic Control

Inspired by the hippocampus mechanism observed in mammals, episodic control-based methods efficiently store and leverage past similar experiences for the processing of new tasks(Lengyel & Dayan, 2007). These methods have found application in model-free Deep Reinforcement Learning (DRL) tasks, particularly for retrieving episodic memory-based state values(Blundell et al., 2016; Pritzel et al., 2017) to address sample inefficiency. Developing episodic control-based methods encompasses various aspects, including state clustering, storage structure, and memory usage. In the realm of state clustering methods, prior approaches typically employ distance-based measurements, such as k-nearest neighbors (KNN)(Kuznetsov & Filchenkov, 2021), model-based state representation(Le et al., 2021), and exact-matching methods (Lin et al., 2018; Li et al., 2023). Concerning state storage structure, in addition to non-parametric memory (Zhu* et al., 2020; Zhang et al., 2019; Pinto, 2020), Hu et al. (2021).proposed a generalized episodic memory with neural networks. Regarding memory usage, most existing methods combine memory values with a joint objective function (Hansen et al., 2018), while other works focus on utilizing memory values as intrinsic rewards (Li et al., 2023; Na et al., 2024; Zheng et al., 2021). Differently, we employ the episodic control mechanism to rectify the credit assignment in MARL algorithms.

6 Limitation and Discussion

While ECA significantly enhances the sample efficiency of existing cooperative MARL algorithms, we acknowledge that there are potential concerns and limitations that need to be considered.

By basing the computation of individual credits on global team returns, ECA effectively addresses concerns about potential non-stationarity. This adaptability ensures the resilience of the learning process in dynamic and non-stationary environments, fostering optimism about ECA's potential. Nevertheless, our proposed ECA approximates the agent credits and does not guarantee convergence to the ground truth reward. Addressing this limitation and exploring the theoretical foundations for such convergence could be a direction for future work.

ECA builds upon the CTDE framework, which aims to resolve cooperative multi-agent tasks. we only evaluate ECA's effectiveness on tasks suitable for such tasks and frameworks. ECA might not perform well on tasks that CTDE cannot handle properly, such as decentralized training and decentralized execution (DTDE) framework (Jiang & Lu, 2022; Tan, 1993; Tampuu et al., 2017; Palmer, 2020). However, the problems and algorithm paradigm that ECA targets and studies are currently crucial in the cooperative MARL field, making ECA a promising method.

7 Conclusion and Future Work

In conclusion, this paper introduces ECA, a simple episodic control-based method for sample-efficient MARL policy optimization. ECA improves the sample efficiency of existing MARL algorithms by directly measuring the agent contribution to the collective success, using the measurement to correct individual action values, and efficiently minimizing the value estimation gap between current and optimal policies. Experimental evaluations conducted on diverse SMAC and GRF tasks have demonstrated the efficacy of ECA. Comparisons with state-of-the-art MARL algorithms and episodic control-based methods have validated the effectiveness of our credit assignment mechanism.

Our future research directions involve (1) improving the scalability and adaptability of ECA to more complex and dynamic environments, (2) applying ECA to more real-world multi-agent scenarios, (3) exploring more promising credit assignment methods to promote the large-scale deployment of existing MARL algorithms, and (4) targeting on safe and reliable MARL algorithms with more novel state measurements.

References

Alaa Awad Abdellatif, Naram Mhaisen, Zina Chkirbene, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Reinforcement learning for intelligent healthcare systems: A comprehensive survey. *arXiv preprint arXiv:2108.04087*, 2021.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control. *arXiv preprint arXiv:1606.04460*, 2016.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=H1lJJnR5Ym`.

Wubing Chen, Wenbin Li, Xiao Liu, Shangdong Yang, and Yang Gao. Learning explicit credit assignment for cooperative multi-agent reinforcement learning via polarization policy gradient. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i10.26364. URL `https://doi.org/10.1609/aaai.v37i10.26364`.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*, pp. 66–83. Springer, 2017.

Steven Hansen, Alexander Pritzel, Pablo Sprechmann, André Barreto, and Charles Blundell. Fast deep reinforcement learning using online adjustments from the past. *Advances in Neural Information Processing Systems*, 31, 2018.

Hao Hu, Jianing Ye, Guangxiang Zhu, Zhizhou Ren, and Chongjie Zhang. Generalizable episodic memory for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 4380–4390. PMLR, 2021.

Jiechuan Jiang and Zongqing Lu. I2q: A fully decentralized q-learning algorithm. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=xdZs1kf-va.

Kun Jiang, Wenzhang Liu, Yuanda Wang, Lu Dong, and Changyin Sun. Credit assignment in heterogeneous multi-agent reinforcement learning for fully cooperative tasks. *Applied Intelligence*, 53(23): 29205–29222, oct 2023. ISSN 0924-669X. doi: 10.1007/s10489-023-04866-0. URL https://doi.org/10.1007/s10489-023-04866-0.

Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Google research football: A novel reinforcement learning environment. *CoRR*, abs/1907.11180, 2019. URL http://arxiv.org/abs/1907.11180.

Igor Kuznetsov and Andrey Filchenkov. Solving continuous control with episodic memory. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2651–2657, 2021.

Hung Le, Thommen Karimpanal George, Majid Abdolshah, Truyen Tran, and Svetha Venkatesh. Model-based episodic memory induces dynamic hybrid controls. *Advances in Neural Information Processing Systems*, 34:30313–30325, 2021.

Máté Lengyel and Peter Dayan. Hippocampal contributions to control: the third way. *Advances in neural information processing systems*, 20, 2007.

Zhuo Li, Derui Zhu, Yujing Hu, Xiaofei Xie, Lei Ma, Yan Zheng, Yan Song, Yingfeng Chen, and Jianjun Zhao. Neural episodic control with state abstraction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=C2fsSj3ZGiU.

Zichuan Lin, Tianqi Zhao, Guangwen Yang, and Lintao Zhang. Episodic memory deep q-networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 2433–2439, 2018.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Hyungho Na, Yunkyeong Seo, and Il chul Moon. Efficient episodic memory utilization of cooperative multi-agent reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LjivA1SLZ6.

Andrew Y Ng, Daishi Harada, and Stuart J Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999.

Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Credit assignment for collective multiagent rl with global rewards. *Advances in neural information processing systems*, 31, 2018.

Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer Publishing Company, Incorporated, 1st edition, 2016. ISBN 3319289276.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Gregory Palmer. *Independent learning approaches: Overcoming multi-agent learning pathologies in team-games*. The University of Liverpool (United Kingdom), 2020.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Boehmer, and Shimon Whiteson. FACMAC: Factored multi-agent centralised policy gradients. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=wZYWwJvkneF.

Rafael Pinto. Model-free episodic control with state aggregation. *arXiv preprint arXiv:2008.09685*, 2020.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International conference on machine learning*, pp. 2827–2836. PMLR, 2017.

Scott Proper and Kagan Tumer. Modeling difference rewards for multiagent learning. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pp. 1397–1398, 2012.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Jianyu Su, Stephen Adams, and Peter Beling. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11352–11360, 2021.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4):e0172395, 2017.

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

Kagan Tumer and Adrian Agogino. Distributed agent-based air traffic flow management. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 1–8, 2007.

Kagan Tumer, Adrian K Agogino, and David H Wolpert. Learning sequences of actions in collectives of autonomous agents. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems: Part 1*, pp. 378–385, 2002.

Binyu Wang, Zhe Liu, Qingbiao Li, and Amanda Prorok. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4): 6932–6939, 2020a.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=Rcmk0xxIQV.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 7285–7292, Apr 2020b. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6220. URL `http://dx.doi.org/10.1609/aaai.v34i05.6220`.

Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. {DOP}: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*, 2021b. URL `https://openreview.net/forum?id=6FqKiVAdI3Y`.

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

David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(02n03):265–279, 2001.

Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021a.

Tianhao Zhang, Yueheng Li, Chen Wang, Guangming Xie, and Zongqing Lu. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. In *International conference on machine learning*, pp. 12491–12500. PMLR, 2021b.

Zhizheng Zhang, Jiale Chen, Zhibo Chen, and Weiping Li. Asynchronous episodic deep deterministic policy gradient: Toward continuous control in computationally complex environments. *IEEE transactions on cybernetics*, 51(2):604–613, 2019.

Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:11853–11864, 2020.

Ziyuan Zhou, Guanjun Liu, and Ying Tang. Multi-agent reinforcement learning: Methods, applications, visionary prospects, and challenges. *arXiv preprint arXiv:2305.10091*, 2023.

Guangxiang Zhu*, Zichuan Lin*, Guangwen Yang, and Chongjie Zhang. Episodic reinforcement learning with associative memory. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HkxjqxBYDB`.