Sequential Cooperative Multi-Agent Reinforcement Learning

Yifan Zang
Institute of Automation, Chinese Academy of Sciences
Beijing, China
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
zangyifan2019@ia.ac.cn

Jinmin He
Institute of Automation, Chinese Academy of Sciences
Beijing, China
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
hejinmin2021@ia.ac.cn

Kai Li
Institute of Automation, Chinese Academy of Sciences
Beijing, China
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
kai.li@ia.ac.cn

Haobo Fu
Tencent AI Lab
Shenzhen, China
haobofu@tencent.com

Qiang Fu
Tencent AI Lab
Shenzhen, China
leonfu@tencent.com

Junliang Xing
Tsinghua University
Beijing, China
jlxing@tsinghua.edu.cn

ABSTRACT

Cooperative multi-agent reinforcement learning (MARL) aims to coordinate the actions of multiple agents via a shared team reward. The complex interactions among agents make this problem extremely difficult. The mainstream of MARL methods often implicitly learn an inexplicable value decomposition from the shared reward into individual utilities, failing to give insights into how well each agent acts and lacking direct policy optimization guidance. This paper presents a sequential MARL framework that factorizes and simplifies the complex interaction analysis into a sequential evaluation process for more effective and efficient learning. We explicitly formulate this factorization via a novel sequential advantage function to evaluate each agent's actions, which achieves an explicable credit assignment and substantially facilitates policy optimization. We realize the sequential credit assignment (SeCA) by dynamically adjusting the sequence in light of agents' contributions to the team. Extensive experimental validations on a challenging set of StarCraft II micromanagement tasks verify SeCA's effectiveness.

KEYWORDS

Cooperative Multi-Agent Reinforcement Learning; Sequential Credit Assignment; Sequential Evaluation

ACM Reference Format:
Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, and Junliang Xing. 2023. Sequential Cooperative Multi-Agent Reinforcement Learning. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) aims to coordinate multiple agents' actions through shared team rewards, which applies to numerous tasks such as robot swarm control [9], autonomous vehicle coordination [1], and network routing [38].

One natural way to address the cooperative MARL problem is the *centralized* approach, which treats the team as a single actor with a joint action space. Although we can trivially apply single-agent RL algorithms to this setting, it usually does not scale well because the joint action space grows exponentially with the agent number [6, 7, 18]. Besides, it is not applicable in real-world settings due to the inherent constraints on agent observability and communication. An alternative approach is to learn *decentralized* policies [3, 27, 39] by independently training agents based on their local observations, but simultaneous exploration brings non-stationarity that causes unstable learning and convergence difficulties [8, 40]. As a result, most works follow the *centralized training with decentralized execution* (CTDE) paradigm [10, 17], where decentralized policies can access extra global information during training.

A crucial challenge of the CTDE paradigm is correctly attributing the global reward from the environment to the agents' individual actions, also known as the *credit assignment problem* [2]. Existing popular MARL frameworks often directly represent the global Q-value as an aggregation of each agent's local value in an inexplicable manner [13, 20, 25, 37]. In this way, these implicit methods avoid explicit coordination analysis and instead fit the complex interactions by neural networks. Although credit assignment may not require an explicit formulation if the policy gradient derived from the centralized critic carries sufficient information [41], it is difficult for decentralized actors to extract direct and valuable knowledge from the implicit information. Without explicitly evaluating each agent's action, they fail to give valuable insights into how well each agent acts and lack direct guidance for policy optimization. In addition, implicit methods often face limitations in expressiveness as they often impose specific constraints on the mixing network [20, 25]. Although some following works [19, 22] attempt to solve this problem, they often introduce extra intractable computations, leading to mediocre performances in complex environments.

To address these problems, we propose a sequential MARL evaluation framework to evaluate each agent successively and explicitly. This framework factorizes the analysis of the complex interactions into a sequential evaluation process. In this factorization, we carry out credit assignment by evaluating agents in a particular sequence,

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

where the evaluation of each agent is based on its preceding agents' actions. We introduce a sequential advantage function under this framework to explicitly formulate the evaluation and optimize the agents' policies in terms of the sequential advantage function. Our sequential credit assignment, referred to as SeCA, equips with a sequence adjustment algorithm and dynamically learns the evaluation sequence according to each agent's contribution to the team.

SeCA avoids the above *inexplicit and difficult* learning and pursues efficient MARL by providing the agents with *explicit and direct* guidance for policy optimization. Our explicable credit assignment is reflected in two aspects: 1) SeCA explicitly evaluates all agent actions and elucidates how well each agent acts, unlike the common practice that decomposes the team reward into individual utilities as an implicit analysis, and 2) the learned evaluation sequence is explicable, which helps the agents collaborate methodically (*c.f.* the fourth part in Section 4.2). The directness of our learning manifests in the explicit formulation of the sequential advantage that directly facilitates policy learning. SeCA also achieves higher expressiveness than most value-decomposition methods, as our centralized critic has no inherent constraints. Although a few works [5, 29] also attempt to give explicit credit assignment, they often perform poorly in complex environments due to simple implementation or strict restrictions. SeCA introduces a more accurate and general evaluation formulation and thus achieves much better performance.

We summarize our main contributions as follows:

(1) We propose *a sequential MARL evaluation framework* that factorizes and simplifies the complex cooperation analysis among agents into a sequence of accessible evaluations.
(2) We formulate the sequential evaluation by introducing a *sequential advantage function* that realizes an explicable credit assignment. In addition, we further provide the upper bound of the proposed sequential advantage's variance.
(3) We present *a sequence adjustment algorithm* to alleviate the impacts caused by the evaluation order. It leverages integrated gradients to dynamically learn the explicable evaluation sequence in light of each agent's contribution.

With these innovations, SeCA enables efficient learning through explicable and direct guidance and achieves competitive performances on a challenging set of StarCraft II micromanagement tasks [21].

2 RELATED WORK

Cooperative MARL coordinates multiple agents by team rewards. The key to promoting coordination is correctly assigning this global reward to each agent, known as the *credit assignment problem.*

The popular *implicit credit assignment* methods often learn a value decomposition from the team reward into individual values, lacking explicable and direct guidance for policy optimization. The earlier work, VDN [25], equips a linear decomposition and ignores the state information. QMIX [20] learns a non-linear mixing network with the global state and maps the individual state-action values into the joint Q-value estimate. Although performing well in various environments, QMIX still faces the mixing network's monotonicity constraint limitation. QTRAN [22] further avoids this representation limitation by using linear constraints between individual utilities and the global state-action value. It guarantees optimal decentralization, but its constraints are computationally intractable, and the relaxations often lead to unsatisfactory performances. VMIX [23] combines A2C with QMIX to extend the monotonicity constraint to value networks and replaces the value network with the monotonic mixing network. QPLEX [28] decomposes Q-values following the dueling structure, transferring the monotonicity condition from Q-values to advantage values. QPD [36] uses integrated gradient attribution to decompose team rewards along trajectory paths. However, whether QPD's individual rewards should be linearly correlated to an agent's contribution remains unclear. Policy-based method, LICA [41], learns end-to-end differentiable policy optimization to remove the monotonicity constraint.

As for the *explicit credit assignment* methods, a few attempts attribute the global reward to individual actions following explicit formulations. Although explicit methods reveal which agent actions are responsible for the team reward, existing works perform poorly as the interactions between agents are highly complex. The notable COMA [5] utilizes a counterfactual baseline to calculate the advantage function. However, its biased advantage evaluates each agent's action based on other agents' current behaviors and ignores their interactions. SQDDPG [29] and Shapley Counterfactual Credits (SCC) [11] distribute the global reward by Shapley Q-value and reflect each agent's marginal contribution through a network or counterfactual method. SQDDPG provides a theoretically justified framework, but the assumption of observability and convex game limits the scope of its application.

3 METHOD

Notations. This paper mainly focuses on a cooperative task with n agents $\mathcal{A} = \{a_1, \ldots, a_n\}$ as a Dec-POMDP [16] defined by a tuple $G =< S, U, P, r, Z, O, n, \gamma >$. We denote joint quantities over agents in bold and joint quantities over agents other than a given agent a with the superscript $-a$. The environment has a true *state* $s \in S$. Each agent a chooses an *action* u_t^a from its action space U^a at timestep t and forms a joint action $\mathbf{u}_t \in (U^1 \times \cdots \times U^n) \equiv U$ that induces a transition according to the *state transition function* $P(s_{t+1}|s_t, \mathbf{u}_t) : S \times U \times S \to [0, 1]$. The *reward function* $r(s, \mathbf{u}) : S \times U \to \mathbb{R}$ yields a global reward, and $\gamma \in [0, 1)$ is the discount factor. We consider partially observable scenarios where agent a acquires its local *observation* $z^a \in Z$ drawn from $O(s_t, a) : S \times \mathcal{A} \to Z$. Each agent has an *action-observation history* $\tau^a \in T^a \equiv (Z \times U^a)^*$ on which it conditions a *policy* $\pi^a(u^a|\tau^a) : T^a \times U^a \to [0, 1]$. The joint policy $\boldsymbol{\pi}$ induces a joint action value function $Q^{\boldsymbol{\pi}}(s_t, \mathbf{u}_t) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} | s_t, \mathbf{u}_t\right]$. Our final goal is to find the optimal joint policy $\boldsymbol{\pi}^*$ such that $Q^{\boldsymbol{\pi}^*}(s_t, \mathbf{u}_t) \geq Q^{\boldsymbol{\pi}}(s_t, \mathbf{u}_t)$ for all $\boldsymbol{\pi}$ and $(s_t, \mathbf{u}_t)$.

3.1 Sequential MARL Evaluation Framework

The interactions in a multi-agent system are complicated. Every agent makes decisions based on the environment interfered with by the other agents, and all agents' actions jointly determine the reward. Therefore, explicitly evaluating each agent's action requires considering the behaviors of other agents. It is hard to determine the impact an agent's action has on the team when we have not assessed other agents. Accordingly, we aim to propose a sequential process to explicitly evaluate each agent's actions one by one and promote cooperation between them.

This section presents a sequential MARL evaluation framework to factorize the complex interaction analysis into a sequence of accessible evaluations. *Our key assumption is that the evaluations of some agents in a team are less affected by other agents' actions.* For instance, when evaluating the action of a staff in a company, the CEO's decision is vital because we have to judge whether the staff obeys the command. On the contrary, a staff's actions intuitively have little impact on evaluating the CEO's decision. When assessing the CEO, we often consider external factors like the market situation modeled as state information s. With this insight, when evaluating each agent's action in a multi-agent system, we can first evaluate the less-affected agents (like the CEO in this example) and then analyze the other agents based on the actions of these already-studied agents. To formulate this sequential evaluation process, we introduce $\texttt{Eval}(\mathcal{A})$ as the evaluation of a set of agents $\mathcal{A}$ to model a sequential MARL evaluation framework. Specifically, the cooperation study on a multi-agent system $\mathcal{A}$ with n agents can be factorized into a sequence of sub-evaluations:

$$\texttt{Eval}(\mathcal{A}) \Leftrightarrow \texttt{Eval}(a_1, a_2, \ldots, a_n) \xrightarrow{\text{Factorize}} \texttt{Eval}(a_i)\ \&\ \texttt{Eval}(a_j|a_i)\ \&\ \texttt{Eval}(a_k|a_i, a_j)\ \&\ \cdots\ \&\ \texttt{Eval}(a_m|\mathcal{A}_{-m}), \tag{1}$$

where $\texttt{Eval}(a_m|\mathcal{A}_{-m})$ represents the evaluation of agent a_m conditional on the agents other than a_m that are already evaluated.

Figure 1: Benefits of the sequential evaluation.

We further provide an inspiring example to illustrate the benefits of the proposed sequential evaluation over directly assessing the interacting system. Agents in a cooperative MARL system learn in every iteration to promote better cooperation, *i.e.*, update their policies to work effectively with others. Synchronously evaluating agents' actions may lead to difficulties and inefficiencies in cooperative policy learning, as each agent may update its policy to better cooperate with the others' *current* policies. As shown in Figure 1(left), when evaluated together, two agents attacking different enemies are both guided to update their policy for the cooperative strategy of *focus fire*. Thus, they may simultaneously change their attack targets to cooperate with each other and cause inefficiencies in learning (change from the current white strategy to the new blue dashed strategy). Sequential evaluation, on the other hand, does not consider the action of Agent 2 when evaluating Agent 1 but only judges whether Agent 1 is attacking its nearest enemy (also essential for the *focus fire* strategy). While Agent 2 is evaluated conditional on the action of Agent 1, thus it will then update its policy to swiftly form the cooperative *focus fire* strategy with Agent 1, illustrated by the blue strategy in Figure 1(right).

Under the sequential MARL evaluation framework, we evaluate each agent based on its preceding agents' actions in a particular order. This framework simplifies the dependencies in the analysis by half since we do not have to consider each agent's subsequent agents when evaluating it. It alleviates the problem that it is hard to judge how good an agent's action is when we have yet to evaluated the others. This framework factorizes the complex interaction analysis among agents into a sequence of accessible evaluations and provides a solid groundwork for the explicit evaluation of each agent.

3.2 Sequential Credit Assignment under The Sequential MARL Evaluation Framework

Following the CTDE paradigm, we utilize a centralized critic for each agent network to follow a gradient that is based on an advantage function A estimated from this critic:

$$g = \nabla_{\theta^\pi} \log \pi\left(u|\tau\right) A. \tag{2}$$

The advantage function A for each actor explicitly deduces how that particular agent contributes to the team. Eqn.(2) shows that the advantage value A directly determines the scale of the policy updating at each iteration. An unreasonable advantage value will lead to oscillation and dilatory learning and may cause convergence to the local optimal solution, even in simple scenarios.

The notable explicit credit assignment method, COMA [5], uses a counterfactual baseline inspired by difference rewards [35]. For each agent a, COMA's counterfactual advantage A^a_{cf} compares the Q-value of action u^a to a counterfactual baseline that *only marginalizes out* u^a while keeping $\mathbf{u}^{-a}$ fixed, *i.e.*,:

$$A^a_{cf}(s, \mathbf{u}) = Q\left(s, \left(u^a, \mathbf{u}^{-a}\right)\right) - \sum_{u'^a} \pi^a\left(u'^a|\tau^a\right) \cdot Q\left(s, \left(u'^a, \mathbf{u}^{-a}\right)\right). \tag{3}$$

The second term of the Eqn.(3) indicates that the counterfactual baseline evaluates agent a's action with the precondition that other agents choose action $\mathbf{u}^{-a}$. It ignores potential joint actions $(u^a, \mathbf{u}^{-a\prime})$ with $\mathbf{u}^{-a\prime} \neq \mathbf{u}^{-a}$ that may lead to unexpected results. Thus, the counterfactual advantage still faces training instability and inefficiency. To better evaluate each agent a, a straightforward practice is to consider the influence of all possible action combinations with u^a, computing the expectation of other agents' actions $\mathbf{u}^{-a}$:

$$A^a(s, \mathbf{u}) = \mathbb{E}_{\mathbf{u}^{-a} \sim \boldsymbol{\pi}^{-a}}\left[Q\left(s, \left(u^a, \mathbf{u}^{-a}\right)\right)\right] - \mathbb{E}_{\mathbf{u}^{-a} \sim \boldsymbol{\pi}^{-a}}\left[\sum_{u'^a} \pi^a\left(u'^a|\tau^a\right) \cdot Q\left(s, \left(u'^a, \mathbf{u}^{-a}\right)\right)\right]. \tag{4}$$

However, the expectation of $\mathbf{u}$s in Eqn.(4) will lead to an independent learning scheme, which is prone to non-stationarity [5, 27].

Under the proposed sequential MARL evaluation framework that factorizes the complex coordination study into a sequence of accessible evaluations, each agent's evaluation is based on its preceding agents' actions, and the actions of the subsequent agents do not influence the evaluation. This trait considers the influence of others when evaluating each agent and avoids independent policy updates. With these properties, we propose a sequential advantage function to improve the counterfactual advantage function in Eqn.(3) and the independent practice in Eqn.(4). Concretely, we give a sequential credit assignment (SeCA) for n agents identified by $\{a_1, a_2, ..., a_n\}$ under a specific sequence $\langle a_1, a_2, ..., a_n\rangle$. Similar equations can be drawn from the rest $(n! - 1)$ orders. Here we

denote $\mathbf{u}^{a_{i:j}} = [u^{a_i}, u^{a_{i+1}}, ..., u^{a_j}]$. After evaluating agent a, we assess the subsequent agents based on u^a. When evaluating agent a_i, the advantage functions of its leading agents $a_1, ..., a_{i-1}$ have been deduced, and the sequential advantage of a_i is based on $\mathbf{u}^{a_{1:i-1}}$:

$$
\begin{aligned}
& A_{SeCA}^{a_i}(s, \mathbf{u}) \qquad (5)\\
= & \mathbb{E}_{\mathbf{u}'^{a_{i+1:n}}}\left[Q\left(s, \left(\mathbf{u}^{a_{1:i}}, \mathbf{u}'^{a_{i+1:n}}\right)\right)\right] - \mathbb{E}_{\mathbf{u}'^{a_{i:n}}}\left[Q\left(s, \left(\mathbf{u}^{a_{1:i-1}}, \mathbf{u}'^{a_{i:n}}\right)\right)\right]\\
= & \sum_{u'^{a_{i+1}}} \cdots \sum_{u'^{a_n}} \prod_{j=i+1}^{n} \pi^{a_j}\left(u'^{a_j} | \tau^{a_j}\right) \cdot Q\left(s, \left(\mathbf{u}^{a_{1:i}}, u'^{a_{i+1}}, \cdots, u'^{a_n}\right)\right)\\
& - \sum_{u'^{a_i}} \cdots \sum_{u'^{a_n}} \prod_{j=i}^{n} \pi^{a_j}\left(u'^{a_j} | \tau^{a_j}\right) \cdot Q\left(s, \left(\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \cdots, u'^{a_n}\right)\right).
\end{aligned}
$$

Proposition 1. *The proposed sequential advantage's variance is upper bounded by the variance of the counterfactual advantage.*

Proof. We first rewrite the counterfactual advantage in Eqn.(3) and the proposed sequential advantage in Eqn.(5) for reading and comprehending convenience:

$$
\begin{aligned}
A_{cf}^{a_i}(s, \mathbf{u}) =& Q\Big(s, \big(u^{a_i}, \mathbf{u}^{-a_i}\big)\Big) - \sum_{u'^{a_i}} \pi^{a_i}\left(u'^{a_i} | \tau^{a_i}\right) \cdot Q\Big(s, \big(u'^{a_i}, \mathbf{u}^{-a_i}\big)\Big)\\
=& \mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\left[Q\Big(s, \big(u^{a_i}, \mathbf{u}^{-a_i}\big)\Big) - Q\Big(s, \big(u'^{a_i}, \mathbf{u}^{-a_i}\big)\Big)\right] \qquad (6)
\end{aligned}
$$

$$
\begin{aligned}
A_{SeCA}^{a_i}(s, \mathbf{u}) =& \mathbb{E}_{\mathbf{u}'^{a_{i+1:n}} \sim \boldsymbol{\pi}^{a_{i+1:n}}}\left[Q\Big(s, \big(\mathbf{u}^{a_{1:i}}, \mathbf{u}'^{a_{i+1:n}}\big)\Big)\right]\\
& - \mathbb{E}_{\mathbf{u}'^{a_{i:n}} \sim \boldsymbol{\pi}^{a_{i:n}}}\left[Q\Big(s, \big(\mathbf{u}^{a_{1:i-1}}, \mathbf{u}'^{a_{i:n}}\big)\Big)\right] \qquad (7)\\
=& \mathbb{E}_{\mathbf{u}'^{a_{i+1:n}} \sim \boldsymbol{\pi}^{a_{i+1:n}}}\Big[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\Big[Q\Big(s, \big(\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}}\big)\Big)\\
& - Q\Big(s, \big(\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}\big)\Big)\Big]\Big]
\end{aligned}
$$

Then, for each agent a_i, we have:

$$
\begin{aligned}
& \mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A_{SeCA}^{a_i}(s, \mathbf{u})\right] = \mathbb{E}\left[\left[A_{SeCA}^{a_i}(s, \mathbf{u})\right]^2\right] - \mathbb{E}\left[A_{SeCA}^{a_i}(s, \mathbf{u})\right]^2\\
= & \mathbb{E}_{\mathbf{u}^{a_{1:n}} \sim \boldsymbol{\pi}^{a_{1:n}}}\left[\mathbb{E}_{\mathbf{u}'^{a_{i+1:n}} \sim \boldsymbol{\pi}^{a_{i+1:n}}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\right.\right.\\
& \left.\left.\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right]\right]^2\right]\\
\leq & \mathbb{E}_{\mathbf{u}^{a_{1:n}} \sim \boldsymbol{\pi}^{a_{1:n}}}\left[\mathbb{E}_{\mathbf{u}'^{a_{i+1:n}} \sim \boldsymbol{\pi}^{a_{i+1:n}}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\right.\right.\\
& \left.\left.\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right]^2\right]\right]\\
= & \mathbb{E}_{\mathbf{u}^{a_{1:i}} \sim \boldsymbol{\pi}^{a_{1:i}}, \mathbf{u}'^{a_{i+1:n}} \sim \boldsymbol{\pi}^{a_{i+1:n}}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\right.\\
& \left.\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right]^2\right]\\
= & \mathbb{E}_{\mathbf{u}^{a_{1:n}} \sim \boldsymbol{\pi}^{a_{1:n}}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\left[Q(s, (u^{a_i}, \mathbf{u}^{-a_i})) - Q(s, (u'^{a_i}, \mathbf{u}^{-a_i}))\right]^2\right]\\
= & \mathbb{E}\left[\left[A_{cf}^{a_i}(s, \mathbf{u})\right]^2\right] - \mathbb{E}\left[A_{cf}^{a_i}(s, \mathbf{u})\right]^2 = \mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A_{cf}^{a_i}(s, \mathbf{u})\right] \qquad (8)
\end{aligned}
$$

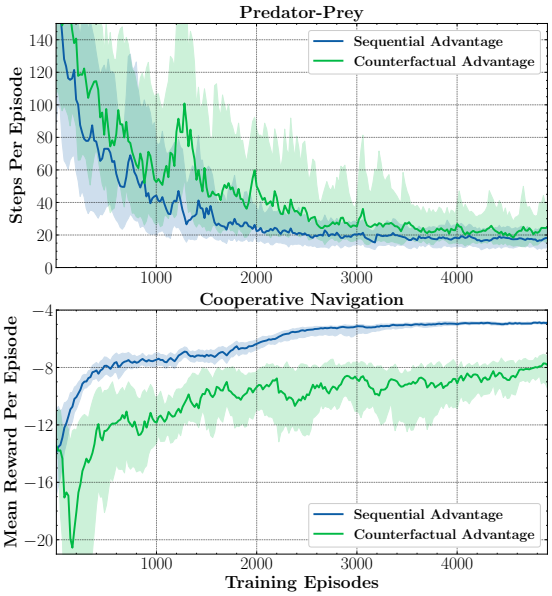

Figure 2: Toy examples: Comparisons between the proposed sequential advantage function and the counterfactual advantage function in two multi-agent particle environments.

Thus, we have $\mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A_{SeCA}^{a_i}(s, \mathbf{u})\right] \leq \mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A_{cf}^{a_i}(s, \mathbf{u})\right]$.

We further analyze the conditions for the tight upper bound. The inequality in Eqn.(8), *i.e.*,

$$
\begin{aligned}
& \mathbb{E}_{u^{a_1} \sim \pi^{a_1}, \ldots, u^{a_n} \sim \pi^{a_n}}\left[\mathbb{E}_{u'^{a_{i+1}} \sim \pi^{a_{i+1}}, \ldots, u'^{a_n} \sim \pi^{a_n}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\right.\right.\\
& \left.\left.\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right]\right]^2\right]\\
\leq & \mathbb{E}_{u^{a_1} \sim \pi^{a_1}, \ldots, u^{a_n} \sim \pi^{a_n}}\left[\mathbb{E}_{u'^{a_{i+1}} \sim \pi^{a_{i+1}}, \ldots, u'^{a_n} \sim \pi^{a_n}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\right.\right. \qquad (9)\\
& \left.\left.\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right]^2\right]\right],
\end{aligned}
$$

is equivalence to Eqn.(10):

$$
\begin{aligned}
& \mathbb{E}_{u'^{a_{i+1}} \sim \pi^{a_{i+1}}, \ldots, u'^{a_n} \sim \pi^{a_n}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\right.\\
& \left.\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right]\right]^2\\
\leq & \mathbb{E}_{u'^{a_{i+1}} \sim \pi^{a_{i+1}}, \ldots, u'^{a_n} \sim \pi^{a_n}}\left[\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\right. \qquad (10)\\
& \left.\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right]^2\right].
\end{aligned}
$$

For reading convenience, we use $\mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}$ to denote $u'^{a_{i+1}} \sim \pi^{a_{i+1}}, \ldots, u'^{a_n} \sim \pi^{a_n}$ and set $\mathbb{E}_{u'^{a_i} \sim \pi^{a_i}}\left[Q(s, (\mathbf{u}^{a_{1:i-1}}, u^{a_i}, \mathbf{u}'^{a_{i+1:n}})) - Q(s, (\mathbf{u}^{a_{1:i-1}}, u'^{a_i}, \mathbf{u}'^{a_{i+1:n}}))\right] = X$. Then we can deduce:

$$
\begin{aligned}
& \mathbb{E}_{\mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}}(X)^2 \leq \mathbb{E}_{\mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}}(X^2)\\
\Leftrightarrow \quad & \mathbb{E}_{\mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}}(X^2) - \mathbb{E}_{\mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}}(X)^2 \geq 0\\
\Leftrightarrow \quad & \mathbb{E}_{\mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}}\left[[X - \mathbb{E}_{\mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}}(X)]^2\right] \geq 0. \qquad (11)
\end{aligned}
$$

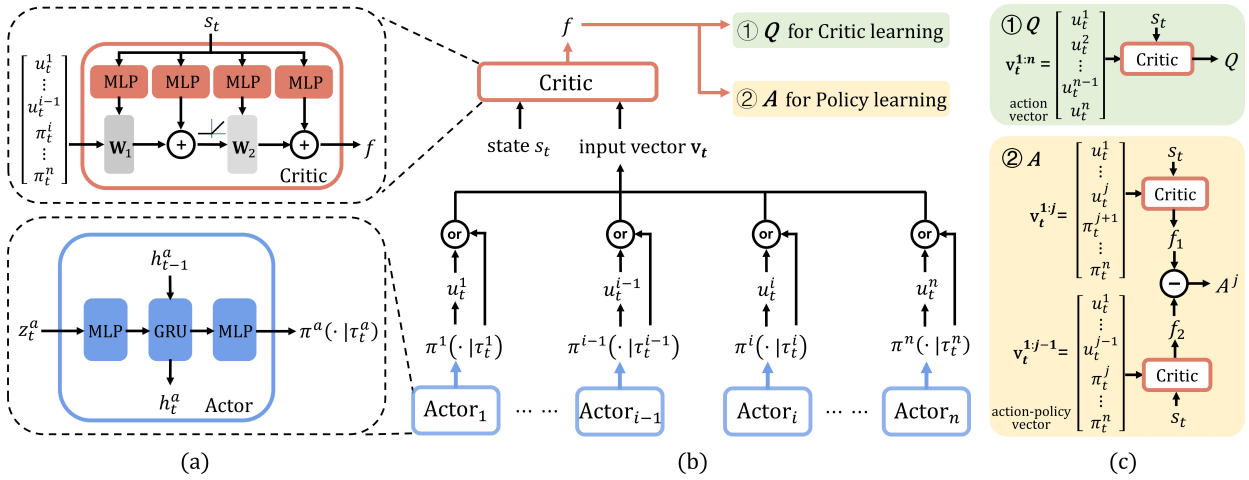

Figure 3: Architecture for SeCA. (a) A centralized mixing critic that maps the state into a set of weights (top) and the agent structure (bottom). (b) The overall SeCA architecture. (c) The critic learning flow (top) and the policy learning flow (bottom).

The equation in (11) holds if and only if $X = \mathbb{E}_{\mathbf{u}'^{i+1:n}\sim\boldsymbol{\pi}'^{i+1:n}}(X)$, $\forall \mathbf{u}'^{i+1:n} \sim \boldsymbol{\pi}'^{i+1:n}$.

Thus, $\mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A^{a_i}_{SeCA}\right] = \mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A^{a_i}_{cf}\right]$ holds only in two cases: (1) $X = \mathbb{E}_{u'^{a_i}\sim\pi^{a_i}}\left[Q(s,(\mathbf{u}^{a_{1:i-1}},u^{a_i},\mathbf{u}'^{a_{i+1:n}})) - Q(s,(\mathbf{u}^{a_{1:i-1}},u'^{a_i},\mathbf{u}'^{a_{i+1:n}}))\right]$ is an constant, $\forall u'^{a_{i+1}} \sim \pi^{a_{i+1}},\ldots,u'^{a_n}\sim\pi^{a_n}$; (2) $\mathbf{u}'^{i+1:n} = \emptyset$, *i.e.*, agent a_i is the last one in the sequence ($i = n$); and $\mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A^{a_i}_{SeCA}\right] < \mathbf{Var}_{\mathbf{u}\sim\boldsymbol{\pi}}\left[A^{a_i}_{cf}\right]$ for other situations. □

To validate our sequential advantage and Proposition 1, we compare the proposed sequential advantage function with the counterfactual advantage in two multi-agent particle environments [12] and follow the environmental settings in [29]. We compare these two advantage functions using the same architecture (COMA's) and only change the way to compute the advantage. Figure 2 illustrates that our sequential advantage helps the agents in *Predator-Prey* capture the prey faster and assists the agents in performing better with significantly smaller variance in *Cooperative Navigation*.

Critic Learning. We train the critic network f_ϕ *on-policy-ly* to estimate the total Q-value and use a variant of TD(λ) [26] adapted for use with neural networks. The critic parameter ϕ is updated by minibatch gradient descent to minimize the following loss function:

$$\mathcal{L}_t(\phi) = \left(y^{(\lambda)}_t - f_\phi(s_t,\mathbf{u}_t)\right)^2, \tag{12}$$

where $y^{(\lambda)}_t = r_t + \gamma\left[\lambda y^{(\lambda)}_{t+1} + (1-\lambda)f_{\phi^-}(s_{t+1},\mathbf{u}_{t+1})\right]$. We utilize a target critic f_{ϕ^-} to improve learning stability [15] and update $\phi^- \leftarrow \phi$ periodically. The top block of Figure 3(c) shows the learning flow of the critic network. The input for critic training is the state s and the action vector $\mathbf{u} = [u^1, u^2, ..., u^n]$ denoted as $\mathbf{v}^{1:n}$.

Policy Learning. Computing each agent's sequential advantage function value $A^{a_i}_{SeCA}$ in Eqn.(5) is extremely time-consuming since a massive amount of Q values with different joint action inputs $\mathbf{u}$ should be calculated. Here we consider an unconventional alternative previously explored in [33, 34, 41] where we directly feed each agent's action distribution parameters (*e.g.*, the action probabilities of a discrete policy or the mean and variance of a Gaussian continuous policy) to estimate the sequential advantage function. We optimize the policy parameter θ by maximizing the following objective, which contains our sequential advantage A^a_{SeCA} in Eqn.(5) and an adaptive entropy regularization term $\mathcal{H}$ [41]:

$$J^a(\theta) = \mathbb{E}_{\tau\sim\pi}\left[\log\pi^a(u^a|\tau^a)A^a_{SeCA}(s,\mathbf{u}) + \mathcal{H}\left(\pi^a(\cdot|\tau^a)\right)\right], \tag{13}$$

where the derivative of the entropy regularization term $\mathcal{H}(\pi^a(\cdot|\tau^a))$ with respect to the i^{th} action probability p^a_i is given by:

$$d\mathcal{H}_i = -\xi\cdot(\log p^a_i + 1)/H(\pi^a(\cdot|\tau^a)), \tag{14}$$

$$\text{and } H(\pi^a(\cdot|\tau^a)) = \mathbb{E}_{u^a\sim\pi^a}\left[-\log\pi^a(u^a|\tau^a)\right]. \tag{15}$$

We share parameters among agents to accelerate learning, and the gradient we use to train the shared agent network is:

$$g = \mathbb{E}_{\tau\sim\pi}\left[\mathbb{E}_a\left[\nabla_{\theta_a}\left(\log\pi^a(u^a|\tau^a)A^a_{SeCA}(s,\mathbf{u}) + \mathcal{H}\left(\pi^a(\cdot|\tau^a)\right)\right)\right]\right]. \tag{16}$$

The policy learning flow is illustrated in the bottom block of Figure 3(c). The inputs of the centralized critic f_ϕ to compute agent a_i's sequential advantage are the state s and two action-policy vectors $\mathbf{v}^{1:i} = [u^1,...,u^i,\pi^{i+1},...,\pi^n]$ and $\mathbf{v}^{1:i-1} = [u^1,...,u^{i-1},\pi^i,...,\pi^n]$. Similarly, the input action-policy vectors to compute the sequential advantage of agent a_{i+1} are $\mathbf{v}^{1:i+1}$ and $\mathbf{v}^{1:i}$.

Under the sequential MARL evaluation framework, the proposed sequential credit assignment explicitly evaluates each agent's action and substantially facilitates policy optimization, generating explicable and direct learning guidance for the agents.

3.3 Dynamic Sequence Adjustment

This section presents one implementation to derive the proper evaluation sequence for SeCA. Our evaluation of each agent a_i in Eqn.(5) is based on its preceding agents' actions $\mathbf{u}^{a_{1:i-1}}$, indicating that agents whose evaluations are grounded in other agents' actions are better placed at the rear of the sequence. Although the CEO-Staff example in Section 3.1 explains the factorization of the cooperation study into a sequential evaluation, roles like CEO and staff are not generalizable to acquire the sequence because multiple agents often

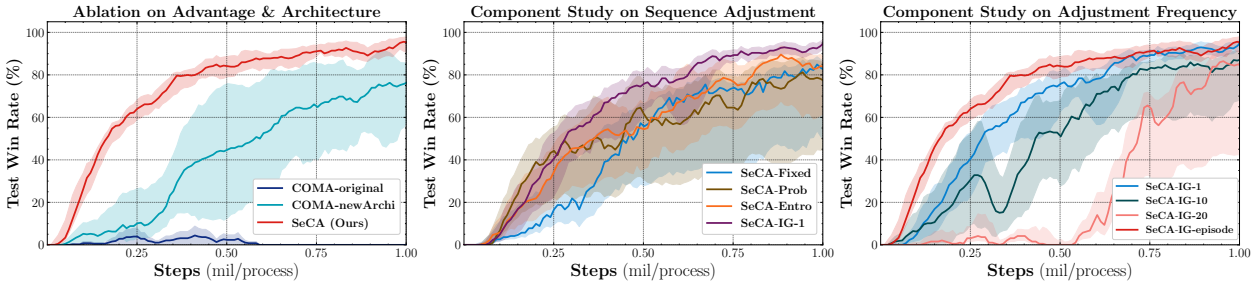

Figure 4: Ablation and component studies on map MMM2 to validate SeCA's critical elements and shows why SeCA works well. (a) verifies the effects of our sequential advantage and network implementation. (b) compares our sequence adjustment method with some other intuitive adjustments. (c) shows the test win percentage with various sequence adjustment frequencies.

play the same role. Therefore, instead of focusing on roles [30, 31], we prefer a universal criterion that fits our sequential advantage.

As a universal element of cooperation, agents' contribution to the team allows a generalizable dynamic sequence learning. The actions of the dominant agents (like the CEO in the example) that contribute more to the team are more correlated with the team reward. Knowing these agents' actions allows a better analysis of whether other agents' actions are beneficial to the team. Therefore, `DescendingOrder`(c^a) realizes a feasible implementation of sequence adjustment, where c^a denotes agent a's contribution.

Integrated Gradients (IG) [24] is a natural tool to deduce contribution in deep learning. It explains how much each input feature of a network F affects the output change along an input path. Given $\alpha \in [0, 1]$ and path function $\tau(\alpha)$ that specifies a path from the baseline $\tau(0) = \vec{b}$ to the input $\tau(1) = \vec{x}$, path integrated gradients along the j^{th} dimension for the input $\vec{x}$ is acquired by:

$$c_j = \text{PathIG}_j^{\tau}(\vec{x}; F) = \int_0^1 \frac{\partial F(\tau(\alpha))}{\partial \tau_j(\alpha)} \frac{\partial \tau_j(\alpha)}{\partial \alpha} \, d\alpha. \tag{17}$$

c_j is the contribution value of x_j to $F(\vec{x}) - F(\vec{b})$, *i.e.*, the difference between prediction $F(\vec{x})$ and the baseline prediction $F(\vec{b})$.

Since we employ a critic network f_ϕ to approximate expected Q-values, its gradient is naturally enabled to extract the contribution of each agent policy π^a to the expected reward obtained from t_1 to t_2, *i.e.*, $(f_\phi^{t_2} - f_\phi^{t_1})$, along a practical integral path. The action-observation history $\tau_{t_1}^{t_2}$ in MARL is a natural candidate for the path τ [36]. Thus, we deduce the contribution during $[t_1, t_2]$ by:

$$c^a = \sum_{j=1}^{|\pi^a|} \text{PathIG}_j^{\tau_{t_1}^{t_2}}(\pi^a; f_\phi), \tag{18}$$

where $|\pi^a|$ gives the number of policy vector's components. We compute each agent's contribution and analyze the agent with a bigger c first to deduce the dynamic sequence. The temporal granularity of the sequence adjustment is studied in the following section. The above contribution-based adjustment only implements a feasible way to adjust the sequence, and the proper adjustment is open for other attempts. We also provide some other intuitive adjustment methods and compare them in Section 4.2.

4 EXPERIMENTS

4.1 Experimental Setup

We consider a challenging set of StarCraft II micromanagement tasks (SMAC) [21] as our experiment environment. The inherent differences among methods and their training procedure (*e.g.*, *on/off-policy learning* for *value-based/policy-based* methods) bring difficulties when comparing methods fairly without introducing additional components (*e.g.*, importance sampling [14, 32] for off-policy methods). To attribute any poor performance of policy-based methods to potential algorithmic limitations or poor training conditions (in particular, *high variance due to small batch sizes or insufficient gradient steps*), we follow [4, 41], training all methods with 32 parallel runners to generate trajectories and using batches of 32 episodes. We evaluate each method every 320K steps with 32 episodes and report the 1st, median, and 3rd quartile win rates across 5 seeds.

4.2 Why SeCA Works Well: Component Studies

Are our sequential advantage function and implementation effective? The sequential advantage function is improved based on COMA's counterfactual advantage, and we have shown our improvement in two toy examples in Figure 2. Afterward, we introduced a f_ϕ approximation and a corresponding network architecture. Here we apply our implementation for COMA's advantage (COMA-newArchi) and compare it with the vanilla COMA and SeCA to show the effects of our sequential advantage and network implementation, respectively. As shown in Figure 4(a), COMA with the vanilla advantage and implementation performs poorly on the *Super Hard* map MMM2 and is significantly improved with our network implementation. The proposed sequential advantage function further accelerates and stabilizes learning.

Does the sequence adjustment algorithm help SeCA perform better? We compare our method with some intuitive adjustments to validate its effects. One could first evaluate agents with higher current-action probability (SeCA-Prob) or lower policy entropy (SeCA-Entro), as they are more confident in their actions. Since SeCA-Prob and Entro get a new order at each step, to be fair, we set the path length in Eqn.(18) to one, *i.e.*, consider agents' contribution based on the transition from s_t to s_{t+1} (SeCA-IG-1). Although formulation (1) suggests that we can assign credit in any sequence, Figure 4(b) illustrates that the variances and learning speeds differ. Both SeCA-Prob and Entro learn faster than a fixed

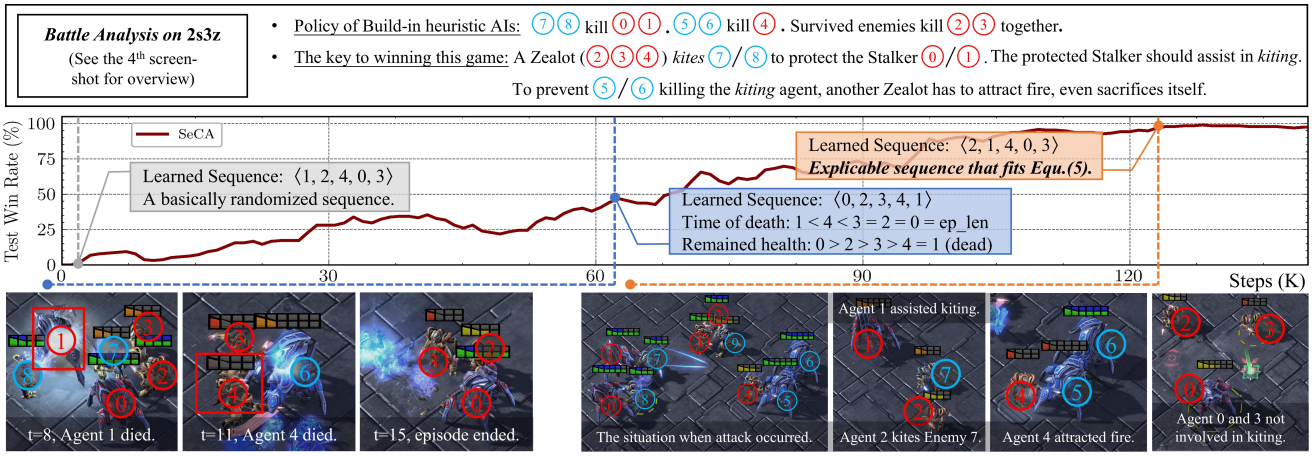

Figure 5: Sequence evolution of SeCA on 2s3z. The learned sequence is explicable and facilitates sophisticated cooperation.

random sequence (SeCA-Fixed), but Prob has a larger variance. Our algorithm performs the best in win rates and stability, while other intuitive adjustments have inferior performance or higher variance.

How did we determine the adjustment frequency? We next study how the sequence adjustment frequency in our method affects the performance. Except per step adjustment (SeCA-IG-1) introduced above, one could also update the sequence after a stage or an episode. If we change the order for every episode (SeCA-IG-episode), $\tau_{t_1}^{t_2}$ in Eqn.(18) represents a whole episode path. As for stage adjustment, it is hard to define a stage in SMAC maps, and the episode length limits vary in diverse maps. Here we set stage lengths to 10 and 20 according to all maps' length limits, denoted as SeCA-IG-10 and IG-20. As Figure 4(c) shows, IG-1 and IG-episode have similar final win rates, but IG-episode converges faster with smaller variance. IG-10 and IG-20's mediocre performance and significant variance may be due to the need for dynamic stage adjustment. *We utilized SeCA-IG-episode for all the other experiments* and will investigate dynamic stage learning in our future work.

Explicable learned sequence. We visualize an illuminating map `2s3z` in Figure 5, demonstrating how the sequence changes and affects the performance as training proceeds. The sequence is adjusted after every episode and is fixed in each battle. In the beginning, the sequence is randomized. At about 60K steps, our approach has learned to adjust the sequence based on survival and health. Although survival and health make sense from some perspectives, they are not proper criteria for deciding an evaluation sequence. As training proceeded, our approach gradually learned explicable sequences that evaluations of agents in the rear should be based on preceding agents' actions. We illustrate this through a battle at around 120K steps. According to our battle analysis, the *kiting* technique is the key to winning this battle. In particular, agents have to make enemy units give chase while maintaining enough distance, so that little or no damage is incurred. In this episode, the dominant Agent 2 that carried out kiting needs Agent 1 to assist in attacking Enemy 7. Agent 4 helped attract Enemies 5 and 6 to ensure the safe kiting execution. Otherwise, Enemies 5 and 6 may attack Agent 2 and stop the kiting. Agents 0 and 3 did not participate in this strategy and carried out side duties. From the contribution perspective, the learned sequence makes sense. Most importantly, it is also explicable to our sequential evaluation that we should evaluate Agents 1 and 4, who assisted Agent 2, based on Agent 2's kiting moves. The auxiliary agents 0 and 3 are evaluated at last based on other agents' actions. The final learned sequence facilitates a sophisticated cooperation strategy. It is also explicable to our sequential credit assignment, *i.e.*, evaluations of agents in the rear of the sequence are based on preceding agents' actions.

4.3 Performance Comparisons

We compare SeCA with prominent baselines in this section to verify SeCA's effectiveness. SeCA compares with COMA and SCC to show its superiority as an *explicit credit assignment method*. COMA is the representative explicit credit assignment method, and SCC is the latest one. SCC is improved based on another explicit baseline SQDDPG and is proved to be better than it; thus, SQDDPG is not involved in our experiments. Besides the explicit credit assignment method, we also choose some notable implicit methods. Among them, LICA is chosen because it is also an on-policy policy-based method. RIIT combines well-known baselines' effective modules and has recently gotten much attention. Thus the comparison with RIIT can fully illustrate the superiority of SeCA as a new credit assignment method. All methods are evaluated on six maps that vary in difficulty by *Easy*, *Hard*, and *Super Hard*. These scenarios involve homogeneous and heterogeneous teams, symmetric and asymmetric battles, allowing a holistic study of all methods.

Existing explicit credit assignment methods often perform poorly in complex environments. However, as illustrated in Figure 6, SeCA demonstrates its superiority and robustness by achieving competitive performances in these scenarios with various characteristics. All methods except two explicit credit assignment methods, COMA and SCC, solve `2s3z` and `1c3s5z`, indicating the poor performance of existing explicit methods. However, SeCA performs the best in convergence speed and stability among all the methods. SeCA's advantage is further extended in map `2c_vs_64zg` and especially `3s5z`. It converges significantly faster than other methods and is the only method that obtains a 100% win rate on `3s5z`. The Zealots in `3s5z` do not purposefully intercept the enemy Zealots, and thus

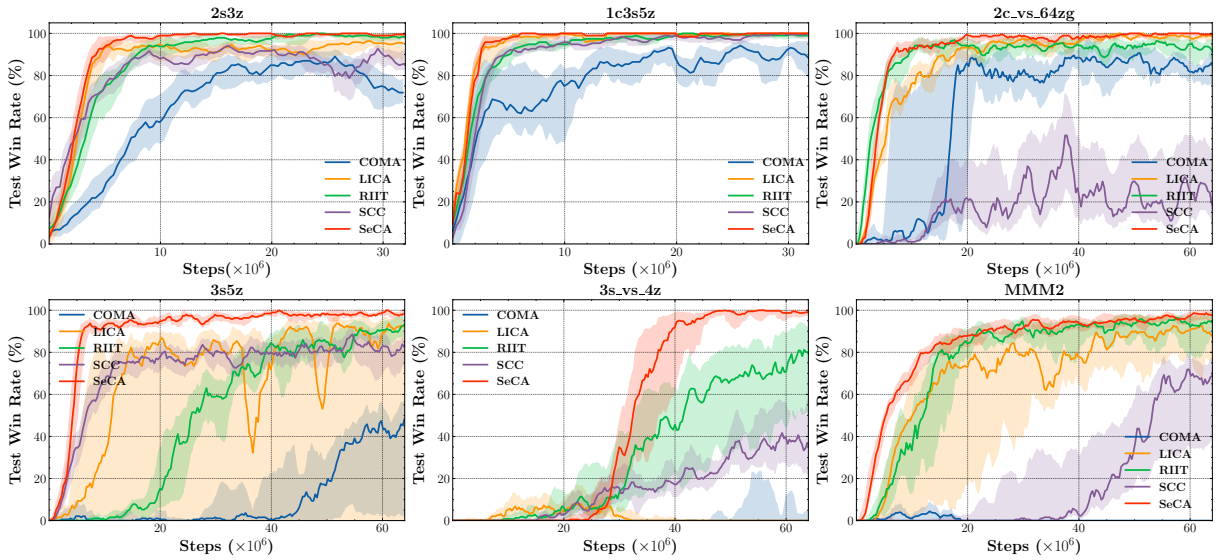

Figure 6: We compare SeCA with representative and prominent baselines on six SMAC scenarios with diverse characteristics to show methods' performances in different styles of environments. All methods are trained in parallel with 32 actors.

the allied Stalkers die very quickly, leading to a guaranteed loss. Therefore, another policy-based method, LICA, has a huge variance in `3s5z`, while SeCA maintains effective learning due to the direct and well-designed guidance for policy optimization. `3s_vs_4z` invalidates COMA and LICA. Stalkers in these maps have to learn to disperse and "kite" enemies. SeCA significantly outperforms all the other methods with the sequential evaluation scheme.

From the experimental results, it is clear that SeCA, as an explicit credit assignment method, obtains higher win rates with more efficient learning than existing baselines, COMA and SCC. SeCA is also a policy-based method, achieving better performance than its counterparts, LICA and RIIT. Notably, RIIT is deemed one of the best-performing policy-based credit assignment methods that incorporate effective modules from several prominent baselines. Thus the above comparisons fully indicate SeCA's superiority.

4.4 Strengths and Limitations Discussion

Although explicit credit assignment methods offer explicable credits and thus give the team direct learning guidance, existing notable methods like COMA often perform poorly due to naive implementation or strict conditions and can hardly win a single battle in complex environments. This situation leads to popular research on value-based implicit credit assignment. While implicit methods have yielded some good results, their learning efficiency is open to further improvement, and their practical guiding significance in the real world is not strong because their inexplicable guidance is not direct and hard to learn. The proposed explicit credit assignment method, SeCA, performs much better than the representative explicit method COMA and the latest explicit method SCC. Besides, SeCA makes full use of good design to give explicable and direct guidance for agents, thus achieving higher efficiency and even better results than the mainstream implicit credit assignment methods. The policy-based method SeCA, with its impressive performance, offers new possibilities for explicit credit assignment and provides a competitive baseline for subsequent credit assignment studies.

Although SeCA is feasible to deal with multi-agent systems with complex interactions theoretically, there may be particular cases where several agents analyzed together would yield more desirable results. In addition, in tasks with plenty of agents, some agents may have no coordination in any way, and it would not be significant to determine a sequence among them. Therefore, we consider integrating coordination graphs into our sequential framework to enhance SeCA. In addition, the contribution-based sequence adjustment is only a possible implementation to inspect sequential credit assignment. More robust adjustments are worthwhile exploring to offer profound viewpoints. As mentioned in Section 4.2, our future interest also includes studying dynamic stage learning to adjust the sequence per stage for adaptive learning.

5 CONCLUSION

This paper presented SeCA, a sequential cooperative MARL framework with explicit credit assignment. We first introduce a sequential MARL evaluation framework and then propose a sequential advantage function for each agent under this framework to realize an explicit credit assignment. We also provide an implementation of sequence adjustment and compare it with other intuitive attempts. SeCA factorizes the complex interaction study among multiple agents into a sequence of accessible evaluations, enabling nontrivial explicit analyses of each agent's action and thus providing direct and explicable guidance for their policy optimization. SeCA dramatically improves the performances of explicit credit assignment methods and achieves higher efficiency and better results than other credit assignment methods. In the future, we will enhance SeCA by exploring new evaluation formulations and sequence adjustments. We believe SeCA will be a new start and a good baseline for subsequent research on the multi-agent credit assignment problem.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under Grant No. 62076238 and 61902402, in part by the National Key Research and Development Program of China under Grant No. 2020AAA0103401, 2022ZD0116401 and 2022ZD0116400, in part by the CCF-Tencent Open Fund, and in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000.

REFERENCES

[1] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. 2012. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* 9, 1 (2012), 427–438.

[2] Yu-han Chang, Tracey Ho, and Leslie Kaelbling. 2004. All learning is local: Multi-agent learning in global reward games. In *Advances in Neural Information Processing Systems*. 808–814.

[3] Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. 2020. Is independent learning all you need in the StarCraft multi-agent challenge? *arXiv preprint arXiv:2011.09533* (2020).

[4] Yali Du, Lei Han, Meng Fang, Ji Liu, Tianhong Dai, and Dacheng Tao. 2019. LIIR: Learning individual intrinsic reward in multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*. 4403–4414.

[5] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*. 2974–2982.

[6] Sven Gronauer and Klaus Diepold. 2021. Multi-agent deep reinforcement learning: A survey. *Artificial Intelligence Review* (2021), 1–49.

[7] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multi-Agent Systems*. 66–83.

[8] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. 2019. A survey and critique of multiagent deep reinforcement learning. In *International Conference on Autonomous Agents and Multi-Agent Systems*. 750–797.

[9] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. 2017. Guided deep reinforcement learning for swarm systems. *arXiv preprint arXiv:1709.06011* (2017).

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

[11] Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. 2021. Shapley counterfactual credits for multi-agent reinforcement learning. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 934–942.

[12] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*. 6382–6393.

[13] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*. 7611–7622.

[14] Rupam Mahmood, Hado van Hasselt, and Richard Sutton. 2014. Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*. 3014–3022.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[16] Frans A Oliehoek and Christopher Amato. 2016. *A concise introduction to decentralized POMDPs*. Springer.

[17] Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. 2008. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research* 32 (2008), 289–353.

[18] Afshin Oroojlooy and Davood Hajinezhad. 2022. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence* (2022), 1–46.

[19] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. 2020. Weighted QMIX: Expanding monotonic value function factorisation. In *Advances in Neural Information Processing Systems*. 10199–10210.

[20] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*. 4295–4304.

[21] 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. 2019. The StarCraft multi-agent challenge. *CoRR* abs/1902.04043 (2019).

[22] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*. 5887–5896.

[23] Jianyu Su, Stephen Adams, and Peter Beling. 2021. Value-decomposition multi-agent actor-critics. In *AAAI Conference on Artificial Intelligence*. 11352–11360.

[24] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*. 3319–3328.

[25] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and Multi-Agent Systems*. 2085–2087.

[26] Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3, 1 (1988), 9–44.

[27] Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *International Conference on Machine Learning*. 330–337.

[28] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2021. QPLEX: Duplex dueling multi-agent Q-learning. In *International Conference on Learning Representations*. 1–9.

[29] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Shapley Q-value: A local reward approach to solve global reward games. In *AAAI Conference on Artificial Intelligence*. 7285–7292.

[30] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. 2020. ROMA: Multi-agent reinforcement learning with emergent roles. In *International Conference on Machine Learning*. 9876–9886.

[31] Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. 2021. RODE: Learning roles to decompose multi-agent tasks. In *International Conference on Learning Representations*. 1–9.

[32] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2017. Sample efficient actor-critic with experience replay. In *International Conference on Learning Representations*. 1–12.

[33] Théophane Weber, Nicolas Heess, Lars Buesing, and David Silver. 2019. Credit assignment techniques in stochastic computation graphs. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2650–2660.

[34] Daan Wierstra and Jürgen Schmidhuber. 2007. Policy gradient critics. In *European Conference on Machine Learning*. Springer, 466–477.

[35] David H Wolpert and Kagan Tumer. 2002. Optimal payoff functions for members of collectives. In *Modeling complexity in economic and social systems*. World Scientific, 355–369.

[36] Yaodong Yang, Jianye Hao, Guangyong Chen, Hongyao Tang, Yingfeng Chen, Yujing Hu, Changjie Fan, and Zhongyu Wei. 2020. Q-value path decomposition for deep multiagent reinforcement learning. In *International Conference on Machine Learning*. 10706–10715.

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

[38] Dayong Ye, Minjie Zhang, and Yun Yang. 2015. A multi-agent framework for packet routing in wireless sensor networks. *Sensors* 15, 5 (2015), 10026–10047.

[39] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*.

[40] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control* (2021), 321–384.

[41] Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. 2020. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*. 11853–11864.