Agent-Temporal Credit Assignment for Optimal Policy Preservation in Sparse Multi-Agent Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

The ability of agents to learn optimal policies is hindered in multi-agent environments where all agents receive a global reward signal sparsely or only at the end of an episode. The delayed nature of these rewards, especially in long-horizon tasks, makes it challenging for agents to evaluate their actions at intermediate time steps. In this paper, we propose Agent-Temporal Reward Redistribution (ATRR), a novel approach to tackle the agent-temporal credit assignment problem by redistributing sparse environment rewards both temporally and at the agent level. ATRR first decomposes the sparse global rewards into rewards for each time step and then calculates agent-specific rewards by determining each agent's relative contribution to these decomposed temporal rewards. We theoretically prove that there exists a redistribution method equivalent to potential-based reward shaping, ensuring that the optimal policy remains unchanged. Empirically, we demonstrate that ATRR stabilizes and expedites the learning process. We also show that ATRR, when used alongside single-agent reinforcement learning algorithms, performs as well as or better than their multi-agent counterparts.

1 Introduction

In cooperative multi-agent reinforcement learning (MARL) multiple autonomous agents learn to interact and collaborate to execute tasks in a shared environment by maximizing a global reward Busoniu et al. (2008). MARL has show considerable potential in solving Dec-POMDPs (Oliehoek & Amato, 2016; Amato, 2024; Zhang et al., 2021) where each agent has access to only local information (partial observation) and need to select actions based on their local action-observation (or sometimes only observation) histories such that they maximize the global (team) reward. Examples of video games where MARL has been applied to such scenarios include StarCraft-II (Vinyals et al., 2019), defence of the ancients (Berner et al., 2019, DOTA), Google football (Kurach et al., 2020), and capture the flag (Jaderberg et al., 2019, CTF). These applications illustrate the potential of MARL to develop sophisticated strategies and behaviors through coordinated teamwork and collaboration.

Despite these successes, cooperative multi-agent systems face the significant challenge of credit assignment, which is crucial for learning effective policies. In the context of multi-agent systems credit assignment has two main aspects: temporal credit assignment and agent credit assignment. Temporal credit assignment involves decomposing sparse, delayed rewards into intermediate time steps within a multi-agent trajectory. Agent credit assignment focuses on discerning the contribution of each agent to these decomposed temporal rewards. Addressing both aspects is crucial for effective learning in cooperative multi-agent systems.

Significant progress has been made to address the credit assignment problem such as (Sunehag et al., 2017; Rashid et al., 2020; Son et al., 2019; Foerster et al., 2018; Freed et al., 2021, VDN, QMIX, QTRAN, COMA, PRD). However, these methods primarily address agent credit assignment and may not be well-suited for environments with sparse or delayed rewards. Moreover, the representations

required for effective credit assignment might not be the same as those needed for learning Q-values or critics. Recent advances have been made to address the issue of temporal credit assignment leverage learning pseudo reward functions (Arjona-Medina et al., 2019; Ren et al., 2021; Liu et al., 2019; Gangwani et al., 2020) in single agent settings and (Xiao et al., 2022; She et al., 2022) in multi-agent settings. These methods attempt to learn a Markovian proxy reward function that replaces the environment's sparse rewards with the learned dense rewards. Motivated by this progress, we aim to address the combined challenge of agent and temporal credit assignment in multi-agent tasks with sparse or delayed rewards via Agent-Temporal Reward Redistribution (ATRR).

In this paper, we aim to address the problem of agent-temporal credit assignment by learning a reward redistribution function that decomposes sparse environment rewards to each time step of the multi-agent trajectory and then further redistribute the temporally decomposed rewards to each agent according to their contribution. We theoretically prove that there exists a class of such reward redistribution functions that can be formulated as potential-based reward shaping (Ng, 1999) under which the optimal policies are preserved in the original reward function of the environment. ATRR extends AREL's (Xiao et al., 2022) reward model that uses a temporal attention module to analyze the influence of state-action tuples on along trajectories followed by agent attention module to identify the relevance of other agents for every agent. This alternation between the two attention modules allows the reward function to identify agent-specific state-action tuples that have key relevance to the sparse environment rewards received by the multi-agent system. Thus, ATRR learns agent-specific temporal rewards, and enables employing single agent reinforcement learning (RL) algorithms like (Tan, 1997; Foerster et al., 2018; Schulman et al., 2017; De Witt et al., 2020, IQL, IAC, IPPO) to solve multi-agent tasks. As a result, we partition the problem of credit assignment from learning Q-functions and critics and leverage the simplicity and scalability of single agent RL algorithms in complex environments.

In summary, our contribution is three folds:-

- We segregate the problem of credit assignment from learning value functions by learning a reward redistribution function that can temporally decompose the sparse environment rewards and assess the contribution of each agent at every time-step.
- We theoretically prove that there exists a family of reward redistribution functions that undertake the potential based reward shaping formulation that preserves the optimal policy in the original reward setting of the environment.
- We empirically validate our approach on 5m_vs_6m battle scenario of SMACLite (Michalski et al., 2023) comparing against various baselines.

2 Related Works

In this section, we will describe several techniques proposed in the past to address the temporal and agent credit assignment problem in single and multi agent systems. Potential based reward shaping is one such method that provided theoretical guarantees of sample-efficient learning of optimal policies in single-agent (Ng, 1999) and multi-agent (Xiaosong Lu, 2011; Devlin & Kudenko, 2011; Devlin et al., 2011) settings.

2.1 Temporal Credit Assignment

Temporal credit assignment deals with the problem of decomposing the sparse or episodic environment rewards into a dense reward function by attributing credit to each time step in the episode. RUDDER (Arjona-Medina et al., 2019) and its variants (Patil et al., 2020) uses contribution analysis to break down episodic rewards to per time-step reward by computing the difference between predicted returns at successive time-steps. A similar line of work proposed in (Zhang et al., 2023) also do return-equivalent contribution analysis. (Liu et al., 2019) leverages auto-regressive architectures

used in natural language processing like Transformers (Vaswani et al., 2017) for attributing credit to every state-action tuple in the trajectory. Both (Efroni et al., 2021; Ren et al., 2021) learn a proxy reward function via a trajectory smoothening based reinforcement learning algorithm by utilizing least squared error. (Harutyunyan et al., 2019) offers a new family of algorithms that uses new information to assign credit in hindsight. (Han et al., 2022) re-desinged the value function to predict the returns for both historical and current steps by approximating these decompositions. (Zhu et al.) introduced a bi-level optimization framework to learn a reward redistribution to learn effective policies. These methods have been developed to learn effective single-agent policies and might not be well suited to MARL settings due to an exponential growth in the joint observation-action space.

In the multi-agent setting, there have been recent works that do temporal credit assignment. IRCR (Gangwani et al., 2020) developed a count-based method to learn a proxy reward function for learning policies for both single and multi-agent settings. AREL introduced in (Xiao et al., 2022) uses attention networks to do return redistribution where as (She et al., 2022) also employs attention encoder network followed by a decode to do agent as well as temporal credit assignment in multi-agent delayed reward settings.

2.2 Agent Credit Assignment

Most of the prior works deal with agent credit assignment in multi-agent systems. (Devlin et al., 2014; Foerster et al., 2018) employ difference rewards methodology to assess the contribution of each agent towards the global reward. Value decomposition networks (Sunehag et al., 2017) decomposed the joint value function of the multi-agent system into agent-specific value functions assuming that they are additive. Subsequent work proposed in (Rashid et al., 2020) introduced monotonicity constraints on the joint Q function to learn individual Q values for each agent. (Son et al., 2019) generalized the approach to decompose joint Q functions to individual Q agent-specific Q functions. (Wang et al., 2020) leverages Shapely values while modeling the joint Q function to do agent credit assignment. (Zhou et al., 2020) propose an entropy regularized actor-critic method to efficiently explore to do multi-agent credit assignment. (Freed et al., 2021) use Transformer attention mechanisms in the critic of an actor-critic method to realize relevant agent subgroups for effective multi-agent credit assignment. The above techniques did not address the problem of temporal credit assignment and hence are inadequate to learn optimal policies in episodic or extremely delayed reward settings.

3 Background

Here we describe our problem formulation as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek & Amato, 2016; Amato, 2024). Subsequently, we describe the episodic multi-agent reinforcement learning setting. Finally, we mathematically show that faulty credit assignment manifests itself in high policy-gradient variance in policy-gradient reinforcement learning algorithms.

3.1 Decentralized Partially Observable Markov Decision Processes

A Dec-POMDP is represented by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{T}, \mathcal{O}, \mathcal{N}, \mathcal{R}_\zeta, \rho_0, \gamma)$ where $s \in \mathcal{S}$ is the environment state space, $a \in \mathcal{A}$ & $\mathcal{A} := \mathcal{A}_1 \times \mathcal{A}_1 ... \mathcal{A}_N$ is the joint action space and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state transition function. $\mathcal{R}_\zeta(r_t|s_t, a_t)$ is the reward function and subsequently, individual agent rewards are sampled according to $r_{1,t}, r_{2,t} ... r_{N,t} \sim \mathcal{R}_\zeta(|s_t, a_t)$. ρ is the initial state distribution and $\gamma \in [0, 1)$ is the discount factor The joint policy of the multi-agent system is denoted by π which is a set of policies, one for each agent, each of which is denoted π_i. Each agent $i \in \{1...N\}$ receives an observation $o_i \in \mathcal{O}_i$ from the observation function $\mathcal{T}(s, i) : \mathcal{S} \times \mathcal{N} \to O$. Because the state is not directly observed, it is typically beneficial for each agent to remember a history of its observations. $\mathcal{H}$ is the set of agent observation (in some cases observation-action) histories up to the current time step t where $h_i = \{a_{i,1}, o_{i,1}, ..., a_{i,t}, o_{i,t}\}$ denote agent i's history. At each time step every agent selects an action $a_i \in \mathcal{A}_i$ according to it's policy $\pi_i : h_i \times \mathcal{A}_i \to [0, 1]$.

$\tau = \{o_{0,1}, a_{0,1}, ...o_{0,N}, a_{0,N}.....o_{|\tau|,1}, a_{|\tau|,1}...o_{|\tau|,N}, a_{|\tau|,N}\}$ is the multi-agent trajectory where $|\tau|$ is the horizon length of the trajectory and $R_{episodic}(\tau) = \sum_{t=0}^{|\tau|}\sum_{i=1}^{N} r_{i,t}(o_{\leq t}, a \leq t)$ is the episodic reward. It is important to make note that the reward that an arbitrary agent i receives at an arbitrary time step t is conditioned only on the past and current observations and actions or rather on the agent histories, $R_{episodic}(\tau) = \sum_{t=1}^{|\tau|}\sum_{i=1}^{N} r_{i,t}(h_{i,t}, h_{\neg i,t})$. In other words, the reward function must be strictly a causal function since actions of the future cannot influence the rewards of the past. The goal of the agents is to determine their individual optimal policies that achieve maximum global return $E_{s_0\sim\rho_0, s\sim\mathcal{P}, a_i\sim\pi_{i..}}(\sum_{t=1}^{|\tau|}\sum_{i=1}^{N}\gamma^t r_{i,t}(h_{i,t},, h_{\neg i,t}) = E_{s_0\sim\rho_0, s\sim\mathcal{P}, a_i\sim\pi_{i..}}(\gamma^{|\tau|}R_{episodic}(\tau))$. Let $r_{global,t}(h_{i,t}, h_{\neg i,t}) = \sum_{i=1}^{N} r_{i,t}(h_{i,t}, h_{\neg i,t})$ which is the temporal reward for time step t of the trajectory.

3.2 Episodic Multi-agent Reinforcement Learning

In most multi-agent systems, every agent receives a reward $r_{global,t}$ after executing joint action a_t in state s_t. However, in episodic MARL setups the agents only receive a feedback from the environment at end of the trajectory called the episodic reward or trajectory return. Thus, the goal of such environments is to maximize trajectory return, which is $E_\tau(R_{episodic}(\tau))$. Such delayed reward settings introduce large bias and variance (Ng, 1999) during the learning process and exacerbate its sample efficiency.

4 Method

4.1 Definition of reward redistribution function

In this paper, we address the challenge of agent and temporal credit assignment in fully cooperative multi-agent systems with episodic global rewards. We want to achieve this by learning a reward redistribution function that preserves the optimal policy of the original reward function of the environment. We define the reward redistribution function conditioned on the multi-agent trajectory τ that decomposes the episodic trajectory reward, aka trajectory return, to each agent based on their contribution to the team's outcome at every time step. Mathematically, we define it as:-

$$R_{episodic}(\tau)(= \sum_{i=1}^{N}\sum_{t=1}^{|\tau|} r_{i,t}(h_{i,t}, h_{\neg i,t})$$

This definition has been adopted by prior works (Xiao et al., 2022; Ren et al., 2021; Efroni et al., 2021) too. The individual rewards functions are strictly causal in nature and can be only conditioned on the current and past observation-actions or histories of the agents.

4.2 Assembling the reward function

We redistribute the trajectory returns temporally to assign credit to each time step in the multi-agent trajectory. Later the temporally redistributed rewards are further decomposed across agents based on their contribution such that Eq 4.1 holds true. Since credit assignment is attributing relative credit to 1) each time step in the multi-agent trajectory followed by 2) each agent at every time step, we can derive the relationship between trajectory return, $R_{episodic}(\tau)$ and the redistributed reward received by agent i at time step t, $r_{i,t}$.

$r_{global,t}$ is the decomposed temporal reward received by the multi-agent system and $r_{i,t}$ is the decomposed agent reward received by redistributing $r_{global,t}$ to every agent at time step t. Thus we can define the relation between them as:

$$\sum_{i=1}^{N} r_{i,t} = r_{global,t}$$

Similarly, based on the definition of the reward redistribution function Eq 4.1

$$\sum_{t=1}^{|\tau|} r_{global,t} = R_{env}$$

Let's define a function W_ω that redistributes the rewards across the temporal axis of the multi-agent trajectory. This function W_ω is parameterized by ω which is a strictly causal function as discussed in Subsection 4.1. Thus, we can express the multi-agent temporal reward at an arbitrary time step t as

$$r_{global,t} = W_{\omega,t} \times R_{episodic}(\tau)$$

Similarly, let's define a function $W_{\omega'}$ that redistributes the temporal rewards at an arbitrary time step t across agents. This function $W_{\omega'}$ is parameterized by ω' which is also a strictly causal function, refer Subsection 4.1. Hence, now we can express the reward that agent i receives as

$$r_{i,t} = W_{\omega',t,i} \times r_{global,t}$$

Finally, deriving the relationship between $r_{i,t}$ and $R_{episodic}(\tau)$ for an arbitrary time-step t

$$r_{i,t} = W_{\omega',t,i} \times W_{\omega,t} \times R_{episodic}(\tau)$$

Here, W_ω is parameterized by $\omega(h_{i,t}, h_{\neg i,t})$ and does temporal credit assignment for the joint multi-agent system and $W_{\omega'}$ which is parameterized by $\omega'(h_{i,t}, h_{\neg i,t})$ attributes temporally assigned credit to each individual agent.

Based on the definition of reward redistribution function

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{t=1}^{|\tau|} r_{i,t} &= R_{episodic}(\tau) \\
(\sum_{i=1}^{N}\sum_{t=1}^{|\tau|} W_{\omega',t,i} \times W_{\omega,t}) \times R_{episodic}(\tau) &= R_{episodic}(\tau) \\
\sum_{i=1}^{N}\sum_{t=1}^{|\tau|} W_{\omega',t,i} \times W_{\omega,t} &= 1 \\
\sum_{t=1}^{|\tau|}(\sum_{i=1}^{N} W_{\omega',t,i}) \times W_{\omega,t} &= 1
\end{aligned}$$

The solution for the above equation is $\sum_{i=1}^{N} W_{\omega',t,i} = 1$ and $\sum_{t=1}^{|\tau|} W_{\omega,t} = 1$.

We now construct the new reward, $\mathcal{R}_{\omega,\omega'}$ to be a function of the original reward $\mathcal{R}$ and the credit $r_{i,t}$ that an arbitrary agent i receives at time step t based on it's relevance towards the multi-agent system's final outcome. This relevance is derived by the reward redistribution function, W_ω and $W_{\omega'}$.

$$\begin{aligned}
R_{\omega,\omega'}(s,a,s') &= R_\zeta(s,a,s') + r_{i,t} \\
R_{\omega,\omega'}(s,a,s') &= R_\zeta(s,a,s') + W_{\omega',t,i} \times W_{\omega,t} \times R_{episodic}(\tau)
\end{aligned}$$

4.3 Optimal Policy Preservation

While the aim is to densify the reward function, we also want to ensure that the optimal policy learned in the new reward function is also optimal in the environment's original reward function. Fortunately, we can show that this is the case:

Proposition 1. *Let's consider two MDPs* $\mathcal{M}_{Env} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{T}, \mathcal{O}, \mathcal{N}, \mathcal{R}_{\zeta}, \rho_0, \gamma)$ *and* $\mathcal{M}_{RRF} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{T}, \mathcal{O}, \mathcal{N}, \mathcal{R}_{\omega,\omega'}, \rho_0, \gamma)$ *as defined in subsection 3.1. The only distinction between* $\mathcal{M}_{Env}$ *and* $\mathcal{M}_{RRF}$ *are the reward functions. If* π^*_θ *is the optimal policy in* $\mathcal{M}_{RRF}$ *then* π^*_θ *is also optimal in* $\mathcal{M}_{Env}$.

Proof. We know that π^*_{theta} is optimal in $\mathcal{M}_{RRF}$. For π^*_{theta} to be optimal in $\mathcal{M}_{Env}$, we need to show that $\mathcal{R}_{\omega,\omega'} = \mathcal{R}_{\zeta} + \mathcal{F}(s, a, s')$ where $\mathcal{F}(s, a, s')$ is a potential based shaping function which is a necessary and sufficient condition for optimal policy preservation (Ng, 1999; Devlin & Kudenko, 2011; Xiaosong Lu, 2011; Devlin et al., 2011). Potential based shaping functions assume the existence of a real valued function $\phi : \mathcal{S} \to \mathbb{R}$ for all $s \to \mathcal{S}$, $a \to \mathcal{A}$ and $s' \to \mathcal{S}$: $\mathcal{F}(s, a, s') = \gamma\phi(s') - \phi(s)$.

It is therefore sufficient to show that the equation 4.2 takes the form $\mathcal{R}_{\omega,\omega'}(s, a, s') = \mathcal{R}_{\zeta}(s, a, s') + \gamma\phi(s') - \phi(s)$. Comparing this format to equation 4.2, assuming $\gamma = 1$ we arrive at $\phi(s') - \phi(s) = W_{\omega',t,i}(h_{i,t}, h_{\neg i,t}) \times W_{\omega,t}(h_{i,t}, h_{\neg i,t}) \times R_{episodic}(\tau)$. This relation holds for $\phi(s) = R_{episodic}(\tau)(\sum_{t=0}^{|\tau|} W_{\omega',t,i}(h_{i,t}, h_{\neg i,t}) \times W_{\omega,t}(h_{i,t}, h_{\neg i,t}))$ □

This result ensures that if the policy π_θ when trained using the reward function $\mathcal{R}_{\omega,\omega'}$ in $\mathcal{M}_{RRF}$ converges to an optimal policy π^*_θ then π^*_θ will also be optimal for the original reward function $\mathcal{R}_{\zeta}$ in $\mathcal{M}_{Env}$.

4.4 Discussion on W_ω and $W_{\omega'}$

In the above subsection we show that $\sum_{i=1}^{N} W_{\omega',t,i} = 1$ and $\sum_{t=1}^{|\tau|} W_{\omega,t} = 1$. W_ω and $W_{\omega'}$ attribute weights to the temporal axis of the multi-agent system and to each agent after temporal credit assignment respectively. These weights represent the distribution of credit across time steps and agents, respectively. It is crucial that these weights are meaningful because if they are not, it can lead to imperfect credit assignment. Imperfect credit assignment occurs when the credit attributed to each agent or each time step does not accurately reflect their true contribution to the final outcome. This mis-attribution can significantly hamper the learning process, leading to sub-optimal policies. For instance, if the weights fail to appropriately highlight the pivotal actions of certain agents at critical time steps, those agents may not receive the necessary feedback to improve their behaviors. Consequently, the overall performance of the multi-agent system may suffer, and the agents may converge to a less effective or even ineffective policy. Therefore, ensuring that these weights are meaningful and accurately capture the contribution of each agent at each time step is vital for effective learning of optimal policy in multi-agent reinforcement learning systems. Hence, the reward redistribution function ($W_\omega(h_{i,t}, h_{\neg i,t})$ and $W_{\omega'}(h_{i,t}, h_{\neg i,t})$) should learn to capture rich representations of the agents' contributions and the temporal dynamics of their actions. This ensures that the weights assigned to each agent and time step are both accurate and meaningful, facilitating effective credit assignment and ultimately leading to the learning of optimal policies.

4.5 Agent temporal reward redistribution (ATRR) architectural details

We merely extend the architecture proposed by Xiao et al. (2022) to not only do decomposition of the episodic reward (trajectory return) temporally but go a step further to decompose them at the agent level. As a result, we encourage the model to predict $r_{i,t}$ and thus, learn implicit temporal and agent weights that adhere to the equation $r_{i,t} = W_{\omega',t,i} \times W_{\omega,t} \times R_{episodic}(\tau)$.

5 Experimental Setup

We demonstrate the effectiveness of our approach ATRR with single-agent and multi-agent reinforcement learning algorithms against some competitive baselines in the 5m_vs_6m battle scenario of the SMACLite (Michalski et al., 2023) environment.

5.1 Baselines

In order to validate the effectiveness of our reward redistribution mechanism we compare its performance with many other forms of reward functions. We train all the baseline reward functions with IPPO(De Witt et al., 2020; Schulman et al., 2017) and MAPPO (Yu et al., 2022) and report them in Fig 1

Episodic rewards: This is the episodic reward setting where each agent receives a global reward signal at the end of the trajectory.

Dense temporal rewards: In this setting, each agent receives the original global dense reward signal described in subsection 5.2.

Dense AREL temporal rewards: This setting employs AREL reward redistribution that temporally assigns rewards to the multi-agent trajectory as described in (Xiao et al., 2022).

Dense IRCR temporal rewards: In this setting, each agent receives a global reward at every time step following this equation $r_{global,t} = R_{episodic}(\tau)/|\tau|$ (Gangwani et al., 2020). This baseline also exemplifies a unique form of reward redistribution in our case where the state and action tuple of each agent is of equal importance and hence each of them receive the same global reward.

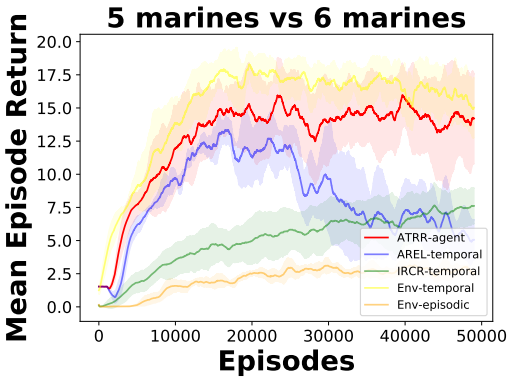
Dense ATRR agent rewards (ours): This is the reward setting proposed in the subsection 4.5.

5.2 Environment

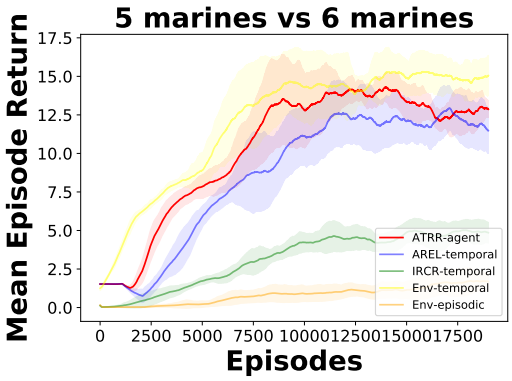
StarCraft Multi-Agent Challenge Lite (SMAClite): The StarCraft Multi-Agent Challenge (SMAC) (Michalski et al., 2023) is an RL environment based on the StarCraft II real-time strategy game, in which a team of agents fights against an opposing team controlled by the game engine's centralized hard-coded AI. We specifically consider the the lightweight and open-source SMACLite version (Michalski et al., 2023). We consider a battle scenario, 5m_vs_6m, where 5 agent-controlled marines battle 6 enemy marines. In this battle situation, the dense reward received by a particular agent while attacking an enemy unit is the difference in the health and shield points removed from that enemy unit in that particular timestep. If a particular agent kills an enemy unit, it receives a reward of 10. Upon defeating the entire enemy team, a reward of (200 / number of agents) is given to each surviving agent. The returns are then normalized such that the maximum possible group return is 20. However, we acummulate the dense reward for each multi-agent trajectory and provide it as a feedback only at the end of the episode.

6 Results and Discussion

As presented in Figure 1, our method ATRR-agent outperform other reward function baselines except for the environment's original dense reward setting as described in subsection 5.2. This particular baseline is an oracle since it has been manually designed to achieve the objective of this specific environment. While training ATRR, we used the same hyperparameters as proposed in (Xiao et al., 2022) with a slight modification to the training procedure. Since AREL (Xiao et al., 2022) was trained with off-policy reinforcement learning algorithms like QMIX (Rashid et al., 2020), they seemed to not require a warm-up period to train the reward function alone. Since in our experiments we train single and multi-agent on-policy policy gradient algorithms, we empirically discovered that a warm-up period (2000 episodes) performed better.



(a) Performance of IPPO in different reward settings.



(b) Performance of MAPPO in different reward settings.

Figure 1: Average agent episodic rewards with standard deviation for task 5m_vs_6m.

7 Conclusion and future work

This paper studied the multi-agent agent-temporal credit assignment problem in MARL tasks with episodic rewards. We proposed a agent-temporal reward redistribution (ATRR) function that theoretically guarantees the preservation of the optimal policy under the original reward function. Our experimental results demonstrate that ATRR outperforms all baselines, showing faster convergence speed.

In future work, we want to explore the agent-temporal reward redistribution by utilizing the attention weights generated by the temporal and agent attention blocks during a forward pass since they naturally fit well in the proposed framework. We want to also demonstrate the effectiveness of our approach against more competitive state-of-the-art baselines and across a variety of other MARL environments of varying difficulty. An interesting line of investigation would be to see the transfer-learning capabilities of such models 1) with more agents than it was trained with 2) across different environments with similar objectives.

References

Christopher Amato. (a partial survey of) decentralized, cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2405.06161*, 2024. 1, 3

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019. 2

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019. 1

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008. doi: 10.1109/TSMCC.2007.913919. 1

Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020. 2, 7

Sam Devlin and Daniel Kudenko. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Adaptive Agents and Multi-Agent Systems*, 2011. URL https://api.semanticscholar.org/CorpusID:1116773. 2, 6

Sam Devlin, Daniel Kudenko, and Marek Grześ. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems*, 14(02):251–278, 2011. 2, 6

Sam Devlin, Logan Yliniemi, Daniel Kudenko, and Kagan Tumer. Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 165–172, 2014. 3

Yonathan Efroni, Nadav Merlis, and Shie Mannor. Reinforcement learning with trajectory feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7288–7295, 2021. 3, 4

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. 1, 2, 3

Benjamin Freed, Aditya Kapoor, Ian Abraham, Jeff Schneider, and Howie Choset. Learning cooperative multi-agent policies with partial reward decoupling. *IEEE Robotics and Automation Letters*, 7(2):890–897, 2021. 1, 3

Tanmay Gangwani, Yuan Zhou, and Jian Peng. Learning guidance rewards with trajectory-space smoothing. *Advances in Neural Information Processing Systems*, 33:822–832, 2020. 2, 3, 7

Beining Han, Zhizhou Ren, Zuofan Wu, Yuan Zhou, and Jian Peng. Off-policy reinforcement learning with delayed rewards. In *International Conference on Machine Learning*, pp. 8280–8303. PMLR, 2022. 3

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019. 3

Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019. 1

Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4501–4510, 2020. 1

Yang Liu, Yunan Luo, Yuanyi Zhong, Xi Chen, Qiang Liu, and Jian Peng. Sequence modeling of temporal credit assignment for episodic reinforcement learning. *arXiv preprint arXiv:1905.13420*, 2019. 2

Adam Michalski, Filippos Christianos, and Stefano V Albrecht. Smaclite: A lightweight environment for multi-agent reinforcement learning. *arXiv preprint arXiv:2305.05566*, 2023. 2, 7

AY Ng. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 278, 1999. 2, 4, 6

Frans A. Oliehoek and Chris Amato. A concise introduction to decentralized pomdps. In *Springer-Briefs in Intelligent Systems*, 2016. URL https://api.semanticscholar.org/CorpusID:3263887. 1, 3

Vihang P Patil, Markus Hofmarcher, Marius-Constantin Dinu, Matthias Dorfer, Patrick M Blies, Johannes Brandstetter, Jose A Arjona-Medina, and Sepp Hochreiter. Align-rudder: Learning from few demonstrations by reward redistribution. *arXiv preprint arXiv:2009.14108*, 2020. 2

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. 1, 3, 7

Zhizhou Ren, Ruihan Guo, Yuan Zhou, and Jian Peng. Learning long-term reward redistribution via randomized return decomposition. *arXiv preprint arXiv:2111.13485*, 2021. 2, 3, 4

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2, 7

Jennifer She, Jayesh K Gupta, and Mykel J Kochenderfer. Agent-time attention for sparse rewards multi-agent reinforcement learning. *arXiv preprint arXiv:2210.17540*, 2022. 2, 3

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019. 1, 3

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. 1, 3

Ming Tan. Multi-agent reinforcement learning: Independent versus cooperative agents. In *International Conference on Machine Learning*, 1997. URL https://api.semanticscholar.org/CorpusID:268857333. 2

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575:350 – 354, 2019. URL https://api.semanticscholar.org/CorpusID:204972004. 1

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7285–7292, 2020. 3

Baicen Xiao, Bhaskar Ramasubramanian, and Radha Poovendran. Agent-temporal attention for reward redistribution in episodic multi-agent reinforcement learning. *arXiv preprint arXiv:2201.04612*, 2022. 2, 3, 4, 6, 7

Sidney N. Givigi Jr Xiaosong Lu, Howard M. Schwartz. Policy invariance under reward transformations for general-sum stochastic games. *Journal of Artificial Intelligence Research*, 41:397–406, 2011. 2, 6

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022. 7

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, 2021. 1

Yudi Zhang, Yali Du, Biwei Huang, Ziyan Wang, Jun Wang, Meng Fang, and Mykola Pechenizkiy. Grd: A generative approach for interpretable reward redistribution in reinforcement learning. *arXiv preprint arXiv:2305.18427*, 2023. 2

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. 3

Tianchen Zhu, Yue Qiu, Haoyi Zhou, and Jianxin Li. Towards long-delayed sparsity: Learning a better transformer through reward redistribution. 3