Lazy Agents: A New Perspective on Solving Sparse Reward Problem in Multi-agent Reinforcement Learning

Boyin Liu [1 2] **Zhiqiang Pu** [1 2 3] **Yi Pan** [2] **Jianqiang Yi** [1 2] **Yanyan Liang** [4] **Du Zhang** [4]

Abstract

Sparse reward remains a valuable and challenging problem in multi-agent reinforcement learning (MARL). This paper addresses this issue from a new perspective, i.e., lazy agents. We empirically illustrate how lazy agents damage learning from both exploration and exploitation. Then, we propose a novel MARL framework called Lazy Agents Avoidance through Influencing External States (LAIES). Firstly, we examine the causes and types of lazy agents in MARL using a causal graph of the interaction between agents and their environment. Then, we mathematically define the concept of fully lazy agents and teams by calculating the causal effect of their actions on external states using the do-calculus process. Based on definitions, we provide two intrinsic rewards to motivate agents, i.e., individual diligence intrinsic motivation (IDI) and collaborative diligence intrinsic motivation (CDI). IDI and CDI employ counterfactual reasoning based on the external states transition model (ESTM) we developed. Empirical results demonstrate that our proposed method achieves state-of-the-art performance on various tasks, including the sparse-reward version of StarCraft multi-agent challenge (SMAC) and Google Research Football (GRF). Our code is open-source and available at https://github.com/liuboyin/LAIES.

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China [2]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China [3]Nanjing AIRIA, Jiangning District, Nanjing, 211100, Jiangsu, China [4]Macau University of Science and Technology, Macau 999078, China. Correspondence to: Zhiqinag Pu <zhiqiang.pu@ia.ac.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

Cooperative multi-agent reinforcement learning (MARL) is an active and fast-growing field that has been applied to a wide range of real-world problems, such as traffic control (Li, 2020), supply chain management (Fuji et al., 2018), network routing (Rosenbaum et al., 2017), and robotics (Perrusquía et al., 2021; Lillicrap et al., 2015).

Centralized training with decentralized execution (CTDE) has been a significant paradigm of cooperative MARL and can effectively deal with nonstationarity while learning decentralized policies for agents (Foerster et al., 2016; Yang et al., 2020; Rashid et al., 2020). In CTDE, agents have access to other agents' information and the global state, but during execution, each agent acts independently based on their policies. Many methods have been proposed following the paradigm of CTDE, including MADDPG (Lowe et al., 2017), MAPPO (Yu et al., 2021), VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2018), MAVEN (Connerney et al., 2015) and QPLEX (Wang et al., 2020a). Among these methods, QMIX stands out as it represents the joint action-value using a non-negative function approximator and achieves superior performance on many MARL benchmarks due to well-designed auxiliary rewards.

Although these MARL approaches progress significantly in complex cooperative tasks, well-designed auxiliary rewards are essential in fostering cooperative or competitive behavior among agents. However, sparse reward scenarios are very typical in MARL applications in the real world. Designing a useful reward function is also notoriously difficult (Abbeel & Ng, 2004), particularly for non-specialists. Sparse reward problem is especially serious when agents must explore large and uncertain environments. In such tasks, agents may not have enough information to develop an optimal behavior and may learn to exploit suboptimal but easily accessible solutions (Sutton & Barto, 2018).

In this paper, we solve the problem of sparse rewards in MARL from a new perspective, i.e., lazy agent. In multi-agent systems, lazy agents refer to agents that do not actively participate in the teamwork or do not contribute significantly to the system's overall performance. Since the reward structure in a sparse reward environment does not adequately

incentivize active participation or contributions to the system's overall performance, agents may lack motivation to contribute (Bolander et al., 2018). As a result, agents may default to passive or lazy behavior, as they do not perceive direct benefits from taking proactive or risky actions.

We propose a novel framework called Lazy Agents Avoidance through Influencing External States (LAIES) to solve sparse reward problems. We first decompose the global states into internal and external states and then construct a causal graph of the interaction between agents and the environment. By analyzing this graph, we identify the causes of lazy agents in MARL and mathematically define the concept of lazy agents and teams. To address lazy agents, we introduce two intrinsic rewards that incentivize agents and a team to have a causal effect on external states. We develope a model for external state transitions to support counterfactual reasoning and use this model to calculate intrinsic rewards. Finally, our approach is evaluated on two scenarios, StarCraft multi-agent challenge (SMAC) (Samvelyan et al., 2019a) and Google Research Football (GRF) (Kurach et al., 2019). Results demonstrate the effectiveness of our method in avoiding lazy agents and improving overall performance in MARL with sparse rewards.

2. Background

This section provides the context necessary to comprehend LAIES and its relationship to existing works.

2.1. Decentralized Partially Observable Markov Decision Process

We consider a fully cooperative multi-agent task as a *decentralized partially observable Markov decision process* (Dec-POMDP) (Oliehoek & Amato, 2016), which can be defined as a tuple $M =< N, S, A, P, r, Z, O, \gamma >$, where N denotes a finite set of agents and $s \in S$ the true state of the environment, $\gamma \in [0, 1)$ the discount factor. At each time step, each agent $j \in N$ receives his own observation $o_j \in O$ and then chooses an action $a_j \in A$ on a global states s, forming a joint action vector $\vec{a}$. It results in a joint reward $r(s, \vec{a})$ and causes a transition on the environment based on the transition function $P(s'|, s, \vec{a})$. Each agent has its own action-observation history $h_j \in T_j \equiv (Z_j \times A)^*$, conditioned by a stochastic policy $\pi_j(a_j|h_j)$. The joint policy π then induces a joint action-value function: $Q^{\pi}_{tot}(s, \vec{a}) = \mathbb{E}_{s_{0:\infty}, a_{0:\infty}} [G_t | s_0 = s, a_0 = \vec{a}, \pi]$, where $G_t = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$ is the expected discounted return.

2.2. Causal Inference

Causal inference is the process of determining whether a change in one variable is the cause of a change in another variable. Understanding causality is important in the context of MARL because it can help agents to better understand how their actions and behaviors are affecting the environment and other agents.

Causal graphs are graphical representations of a set of variables and their causal relationships to one another. In a causal graph, variables are represented as nodes, and the relationships between variables are represented as edges connecting the nodes. The direction of the edge indicates the direction of causality, with the arrow pointing from the cause to the effect (Pearl, 1988).

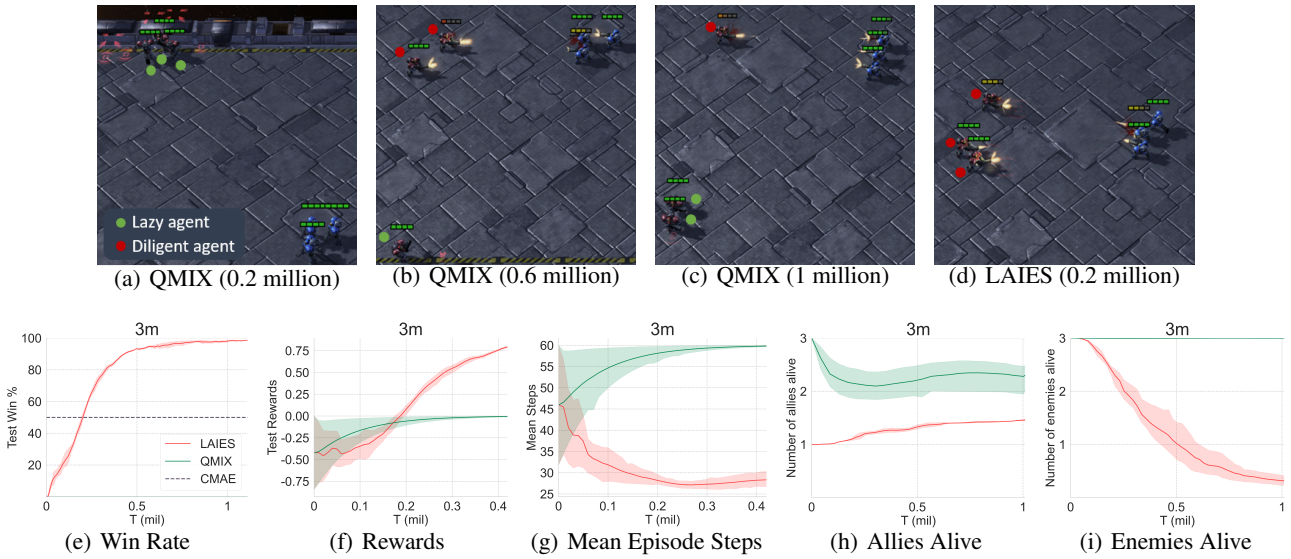
Do-calculus process is a causation framework that allows us to identify causal effects using the causal assumptions encoded in the causal graph. It gives us tools to identify any causal estimand that is identifiable. Concretely, a causal estimand is denoted as $P(O|do(M = m), C = c)$, where O is a set of outcome variables, M is a set of treatment variables, and C is a set of covariates.

2.3. Related Works

Existing works addressing the problem of sparse rewards in MARL from several perspectives.

The most straightforward solution to the sparse reward problem is enhancing exploration. The idea behind this approach is that by increasing the agent's ability to explore the state space, it can more efficiently discover states that lead to higher rewards. MAVEN (Connerney et al., 2015) learns a diverse ensemble of monotonic approximations with the help of a latent space to explore. One standard method for enhancing exploration is using an exploration bonus (Pathak et al., 2017; Liu et al., 2021), an additional reward signal provided to the agent for visiting novel states, such as curiosity-driven (Pathak et al., 2017). Lastly, some methods use an ensemble of agents to enhance exploration, where a group of agents works together to explore the state space more efficiently. CMAE (Liu et al., 2021) shares a common exploring goal selected from multiple projected state spaces. However, these methods only emphasize exploration without investigating which states are worth exploring, resulting in limited performance in complex scenarios.

Encouraging influential behaviors (Liu et al., 2020) between agents is also a proposed approach for addressing the sparse reward problem in MARL. Influential behaviors can be achieved in MARL by directly shaping other agents' policy updates (Foerster et al., 2017; Letcher et al., 2018), maximizing the mutual information (MI) between agents' actions (Jaques et al., 2019), identifying the relationship between its behavior and the other agent's future strategy (Xie et al., 2021), considering each agent's impact on the converged policies of other agents (Kim et al., 2022), and maximizing the MI associated with high-level collaborative behaviors and minimizing the MI with low-level one (Li et al., 2022).



(a) QMIX (0.2 million) (b) QMIX (0.6 million) (c) QMIX (1 million) (d) LAIES (0.2 million)

(e) Win Rate (f) Rewards (g) Mean Episode Steps (h) Allies Alive (i) Enemies Alive

Figure 1. (a), (b) and (c) depict the policies obtained by QMIX after 0.2 million steps, 0.6 million steps, and 1 million steps of training respectively, under the condition of sparse rewards. (d) illustrates the policy obtained by LAIES after 0.2 million steps of training. The green and red points label lazy agents and diligent agents, respectively. The curves in (e) - (i) represent the average test results of five different random seeds.

However, encouraging influential behaviors may lead to unwanted coalitions, where certain groups of agents collaborate to achieve their own goals but not necessarily the overall objective of the task. Additionally, subgoal-based approaches have emerged as an effective method for addressing the sparse reward problem in MARL. HDMARL (Tang et al., 2018) uses a predefined set of subgoals based on domain knowledge, while MASER (Jeon et al., 2022) automatically generates subgoals from experience replay buffers for multiple agents. However, domain-knowledge-based subgoal-based approaches are not easily extended to different tasks, and automatically generated subgoals may not provide adequate performance.

3. Why QMIX Fails with Sparse Reward

In this section, we will use the QMIX method as an example and discuss the challenges many existing MARL methods encounter in sparse reward tasks. Specifically, we will use the 3m task in SMAC to illustrate these challenges. The 3m task involves controlling a team of three red Marines to eliminate three blue Marines, with three types of reward, i.e., +1 for a win, 0 for a tie, and -1 for a loss.

Figures 1(a), 1(b) and 1(c) illustrate that QMIX falls into a local optimum, resulting in a negative strategy of controlling the red Marines to simply survive until the end of the round. At least one agent is an avoider who learns to hide at the edge of the map to avoid being discovered by the enemy. From a cooperative perspective, these avoiders can be considered lazy agents. The existence of these lazy agents will lead to the failure of the whole task. As seen in Figure 1(f), the reward obtained by agents for actively attacking the blue Marines is lower than that obtained by the negative evasion strategy when the win rate is below 50%. Consequently, the structure of the reward promotes agents to engage in laziness.

The presence of lazy agents leads to inadequate exploration. From an exploration perspective, a strategy with a win rate of 50% is harder to explore than an evasion strategy, although the average rewards for both strategies are the same. The evasion strategy only requires the agent to select the move action. Hence only the red Marines' state space is required to be explored to find this strategy. However, a 50% win rate strategy necessitates exploring the entire state space. In Figures 1(a), 1(b) and 1(c), lazy agents hiding at the edge of the map without interacting with the blue Marines leads to low exploration efficiency of the state space of the blue Marines. Figures 1(h) and 1(i) confirm this, demonstrating that QMIX's learned approach continuously maintains a greater survival rate for the red Marines but fails to eliminate the enemy successfully. This means that the lazy behavior of agents leads to the health-related portion of the blue Marines' state space of the blue Marines being insufficiently explored.

Additionally, the presence of lazy agents leads to ineffective cooperation. As depicted in Figure 1(d), the agent requires a coordinated offensive strategy to overcome the opponent. The presence of lazy agents (shown in Figures 1(b) and

1(c)), however, results in the gradual elimination of agents. Figure 1(i) demonstrate that the lazy behavior of the agent causes no blue Marine to be killed during the whole training process.

Hence, QMIX fails due to inadequate exploration and ineffective cooperation caused by lazy agents. CMAE encourages agents to explore the entire state space jointly, thus alleviating the problem of low exploration efficiency caused by the lazy behavior of agents. Therefore, CMAE can learn a strategy with a 50% win rate illustrated in Figure 1(e). LAIES is an effective approach to address the issue of lazy agents by encouraging diligent behavior in agents. As shown in Figures 1(e), 1(g), 1(h), and 1(i), LAIES outperforms other methods in terms of winning rate, task completion speed, proactive attacks, and enemy kills. These experimental results demonstrate that LAIES can effectively overcome the challenges of sparse reward environments and prevent the emergence of lazy agents. In the following sections, we will provide a detailed methodology for LAIES and present further experimental studies to support our findings.

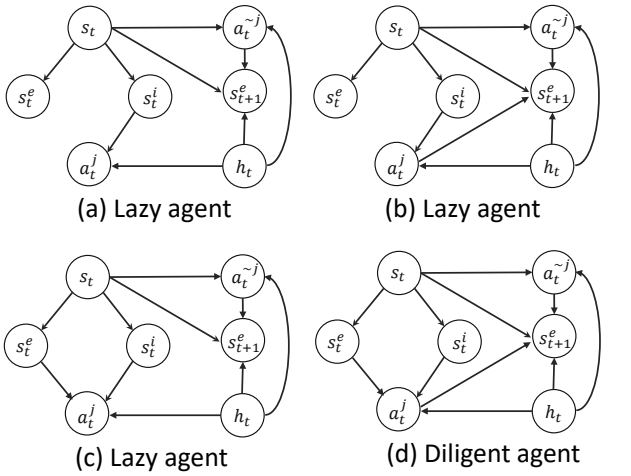


Figure 2. Causal diagrams of three types of lazy agents and diligent agent. We mainly consider agent j here. h_t is the historical state trajectory information. $a_t^{\sim j}$ is joint action except for agent j.

4. Method

This section introduces our framework LAIES to improve cooperative MARL with sparse reward.

4.1. What is the Lazy Agent?

To study lazy agents, the first step is to define lazy agents and teams. Firstly, we define external states:

Definition 4.1. The global states $s_t = (s_t^i, s_t^e)$ can be divided into two parts: those states directly associated with a group of agents, such as position, are internal states s_t^i. Other states that agents can influence that are not directly associated with them is the **external state** s_t^e, such as enemies' positions and health in 3m task.

Remark 4.2. External states may not be present in all tasks, such as matrix games, but they are present in most complicated tasks. This paper's research focuses on tasks involving external states.

In MARL, agents alter their internal states through action selection, leading to changes in external states. The final goal's reach always correlates more with external states, such as three blue Marines' death in the 3m task of SMAC, the ball entering a goal in GRF, and a target point covered in a coverage task. Exploring external states can be more difficult under sparse reward conditions due to the potential for failure and punishment, as illustrated in Figure 1(f) where LAIES receives punishment when its win rate is below 50%. Therefore, this work considers a lazy agent as one whose strategy cannot influence external states.

Figure 2 shows the causal diagrams of three types of lazy agents j and one diligent agent j. In Figure 2(a), external states s_t^e are not a cause of the agent j's action a_t^j which is also not a cause of s_{t+1}^e. For example, in SMAC, an agent may run to the edge of the map without observing and influencing any enemies (shown in Figure 1(a)). In Figure 2(b), s_t^e is not a cause of a_t^j, but a_t^j is a cause of s_{t+1}^e. For example, the agent may adopt a random strategy that has only short-term impacts on external states and does not significantly influence them in the long term. In Figure 2(c), s_t^e is a cause of the agent j's action a_t^j which is not a cause of s_{t+1}^e, such as an agent in SMAC just avoiding an enemy within sight (shown in Figures 1(b) and 1(c)).

In the context of MARL, lazy agents typically refer to the third type of agent described in Figure 2(c). Since external states generally are part of an agent's observation, the agent often uses it to make decisions, but it may not necessarily affect the transition of external states. Emphasizing the influence of the s_t^e on a_t^j makes the agent j more attentive to external states' information. Figure 2(d) illustrates the causal graph for a diligent agent, where the agent's decisions are influenced by and influence external states. Since lazy agents are a common issue with sparse reward, this work aims to avoid their occurrence, i.e., to enhance the agent's influence on external states.

The above analysis qualitatively analyzes the emergence of lazy agents in MARL scenarios. Next, we need to provide a mathematical definition of lazy agents. According to Figure 2(c) and Figure 2(d), the first step is to calculate the causal effect between a_t^j and s_{t+1}^e. As shown in Figure 2(d), s_{t+1}^e have four parents, i.e., a_t^j, h_t, $a_t^{\sim j}$ and s_t, all of which are a direct cause of s_{t+1}^e, which bring both causal and non-causal

associations between a_t^j and s_{t+1}^e. The causal effect of a_t^j on s_{t+1}^e can be calculated as

$$\begin{aligned} Y_{s_{t+1}^e}(A_t^j = a_t^j) &= P(s_{t+1}^e | do(A_t^j = a_t^j)) \\ &= \sum_{w_t} P(s_{t+1}^e | a_t^j, w_t) P(w_t | do(A_t^j = a_t^j)) \\ &= \sum_{w_t} P(s_{t+1}^e | a_t^j, w_t) P(w_t) \end{aligned} \tag{1}$$

where $w_t = \{s_t, h_t, a_t^{\sim j}\}$ is a node set; the potential outcome $Y_{s_{t+1}^e}(A_t^j = a_t^j)$ denotes what your outcome would be, if you were to take treatment $A_t^j = a_t^j$. Since $Y_{s_{t+1}^e}(A_t^j = a_t^j)$ only calculates the causal effect at time t, we have $P(w_t) = 1$. Then,

$$Y_{s_{t+1}^e}(A_t^j = a_t^j) = P(s_{t+1}^e | a_t^j, w_t) \tag{2}$$

Then, the treatment effect of a_t^j to s_{t+1}^e can be mathematically calculated as:

$$\begin{aligned} \tau_{a_t^j} &= D_{KL}\left[Y_{s_{t+1}^e}(A_t^j = a_t^j) \| Y_{s_{t+1}^e}(A_t^j \neq a_t^j) \right] \\ &= D_{KL}\left[P(s_{t+1}^e | a_t^j, w_t) \| P(s_{t+1}^e | w_t) \right], \end{aligned} \tag{3}$$

where $\tau_{a_t^j}$ is called as individual diligence degree (IDD).

Similarly, we can also calculate the treatment effect of joint action $\vec{a}_t$ to s_{t+1}^e:

$$\begin{aligned} \tau_{\vec{a}_t} &= D_{KL}\left[Y_{s_{t+1}^e}(\vec{A}_t = \vec{a}_t) \| Y_{s_{t+1}^e}(\vec{A}_t \neq \vec{a}_t) \right] \\ &= D_{KL}\left[P(s_{t+1}^e | \vec{a}_t, s_t, h_t) \| P(s_{t+1}^e | s_t, h_t) \right], \end{aligned} \tag{4}$$

where $\tau_{\vec{a}_t}$ is called as collaborative diligence degree (CDD) and the potential outcome $Y_{s_{t+1}^e}(\vec{A}_t = \vec{a}_t)$ denotes what your outcome would be, if you were to take treatment $A_t = \vec{a}_t$. With above derivation, we can give the mathematical definition of a fully lazy agent and lazy team:

Definition 4.3. The agent j is a fully lazy agent iff $\sum_{t=0}^{T} \tau_{a_t^j} = 0$.

Definition 4.4. The team is fully lazy iff $\sum_{t=0}^{T} \tau_{\vec{a}_t} = 0$.

Remark 4.5. Since the final objective always correlates with external states, we believe an agent or team is fully lazy when they fail to have any influence on external states within an episode.

4.2. Individual Diligence Intrinsic Motivation

In practice, most agents do not exhibit complete laziness but are less diligent than desired. To encourage more proactive behavior, we can use IDD as intrinsic motivation to encourage agents to be diligent. Using Definition 4.3, the Individual Diligence Intrinsic motivation (IDI) can be calculated as

$$r_t^{IDI} = \sum_{j=1}^{|N|} \tau_{a_t^j}. \tag{5}$$

External States Transition Model (ESTM): The calculation of IDI involves the distribution of external states. However, sampling from the distribution during online exploration could bring big variance. Furthermore, the environment model is not available. To address this problem, we model the transition of external states using neural networks, shown in Figure 3. ESTM uses states and joint action as input to predict external states at next time. Under CTDE paradigm, whole episode's global states are available during the training phase when we use and train ESTM. With ESTM, we can conveniently conduct do-operator on agent's action and calculate a potential outcome.

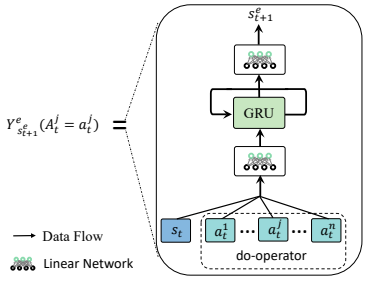


Figure 3. Network architecture of ESTM. GRU is used to capture historical information.

Since the state transition is always fixed given s_t, h_t and $\vec{a}_t$, $P(s_{t+1}^e | do(\vec{a}_t), s_t, h_t)$ and $P(s_{t+1}^e | s_t, h_t)$ obey one-point distribution. Therefore we replace the KL-divergence of distribution with MSE loss of external states to calculate the treatment effect in practice. $\tau_{a_t^j}$ is calculated as:

$$\tau_{a_t^j} = MSE(Y_{s_{t+1}^e}(A_t^j = a_t^j), Y_{s_{t+1}^e}(A_t^j \neq a_t^j)). \tag{6}$$

In Eq.6, $Y_{s_{t+1}^e}(A_t^j = a_t^j)$ is observable while $A_t^j \neq a_t^j$ is counterfactual. To compute the latter, we can intervene on a_t^j by replacing it with a counterfactual action which is used to get a potential counterfactual outcome of s_{t+1}^e with the help of ESTM. We can enumerate all other available actions of the agent and calculate the mean value of these potential counterfactual outcomes as $Y_{s_{t+1}^e}(A_t^j \neq a_t^j)$. Mathematically, it can be formulated as follows:

$$\begin{aligned} \tau_{a_t^j} = MSE(&Y_{s_{t+1}^e}(A_t^j = a_t^j), \\ &\frac{1}{|A|-1} \sum_{k=1, a_k \neq a_t^j}^{|A|} Y_{s_{t+1}^e}(A_t^j = a_k)) \end{aligned} \tag{7}$$

where $|A|$ is the size of action set A.

4.3. Collaborative Diligence Intrinsic Motivation

IDI encourages each agent to exert its influence on external states. However, emphasizing each agent's influence does not mean maximization of the whole team's influence. To address this problem, we propose collaborative diligence intrinsic motivation (CDI), which calculates the causal effect of joint action on external states. Specifically, CDI is calculated as

$$r_t^{CDI} = MSE(Y_{s^e_{t+1}}(\vec{A}_t = \vec{a}_t), \frac{1}{|A|^n - 1} \sum_{k=1, \vec{a}_k \neq \vec{a}_t}^{|A|^n} Y_{s^e_{t+1}}(\vec{A}_t = \vec{a}_k)). \tag{8}$$

To calculate CDI, we need to calculate all possible counterfactual combinations of joint action, which grow exponentially as agents increase. In practice, we can randomly sample several joint actions instead to calculate the mean counterfactual potential outcome.

4.4. Overall Optimization Objective

In this paper, we have introduced two terms of intrinsic rewards to avoid lazy agents in MARL with sparse reward. The ultimate intrinsic reward is calculated as

$$r^I = \beta_1 * r^{IDI} + \beta_2 * r^{CDI}, \tag{9}$$

where β_1 and β_2 are non-negative scaling factors of IDI and CDI, respectively. LAIES adopts QMIX to estimate joint action-values, and is trained in an end-to-end way to minimize the following loss:

$$\mathcal{L}(\theta) = \sum_{i=1}^{b} \left[(y_i^{tot} - Q_{tot}(\tau, a, s; \theta))^2 \right], \tag{10}$$

$$y_i^{tot} = r^E + r^I + \gamma \max_{a'} Q_{tot}(\tau', a', s'; \theta^-), \tag{11}$$

where r^E represents the extrinsic reward, b is the batch size of transitions sampled from the replay buffer, θ^- the parameters of a target network.

5. Experiments

In this section, we evaluate LAIES on a variety of complex multi-agent tasks with sparse reward to answer the following questions:

Q1 Can LAIES effectively avoid lazy agents and outperform related baselines? (Sections 5.2 and 5.3)

Q2 Whether IDI and CDI contribute collectively to the final performance of LAIES? (Section 5.4)

Q3 Whether a more successful strategy results in agents having a greater influence on external states? And whether IDD and CDD effectively reflect the diligence of the agents and the team ? (Section 5.5)

Q4 Whether LAIES is still effective in combining with the policy-gradient-based method? (Appendix B.1)

5.1. Experimental Setup

To evaluate the effectiveness of LAIES, we conduct experiments with different scenarios on two popular MARL benchmarks, i.e., SMAC [1] (Samvelyan et al., 2019a) and GRF (Kurach et al., 2019). We use the sparse reward setting in both games without specific instructions. In SMAC, LAIES is compared against many state-of-the-art (SOTA) methods, including CMAE (Liu et al., 2021), MAVEN (Mahajan et al., 2019), MASER (Jeon et al., 2022), QMIX (Rashid et al., 2018), and RODE (Wang et al., 2020b). Since CMAE did not publicly release their code for the SMAC scenario, we used the highest reported score from their paper as a representation of their performance in the graph. To show the outstanding performance of LAIES more conveniently in SMAC, we have added a curve QMIX-DR (QMIX with dense rewards). In GRF, we compare LAIES with QMIX, QPLEX (Wang et al., 2020a), CDS (Li et al., 2021), and MAVEN. For evaluation, all experiments are carried out with six random seeds. In SMAC, external states refer to opponents' positions and health, whereas opponents' and ball's positions and directions are in GRF. The details of the architecture of our method and baselines can be found in Appendix A.1. More details about SMAC and GRF can be found in Appendix A.2 and A.3, respectively.

5.2. Superior Performance on SMAC

The proposed method demonstrates superior performance compared to the SOTA methods, as evidenced by the results in Figure 4.

2m_vs_1z is an easy map, where the ally has 2 Marines and the enemy has 1 Zealot. Here, external states are observed Zealot's state. In LAIES, Marines are encouraged to influence Zealot's state, including position and health, strengthening the exploration of external states space. CMAE can reach a win rate of over 40% with 0.5 million training steps. As demonstrated in the learning curves of Figure 4(a), our method reaches a higher level of performance than CMAE.

MMM is an easy map where the ally has 1 Medivac, 2 Marauders, and 7 Marines fighting with an enemy team of equal strength. As shown in Figure 4(b), LAIES has a lower variance and converges faster than QMIX-DR. It is worth noting that the experimental results indicate that LAIES can reach nearly 100% success rate in this task.

5m_vs_6m is a hard map where the ally has 5 Marines, and

[1] We use the SC2.4.10 version.

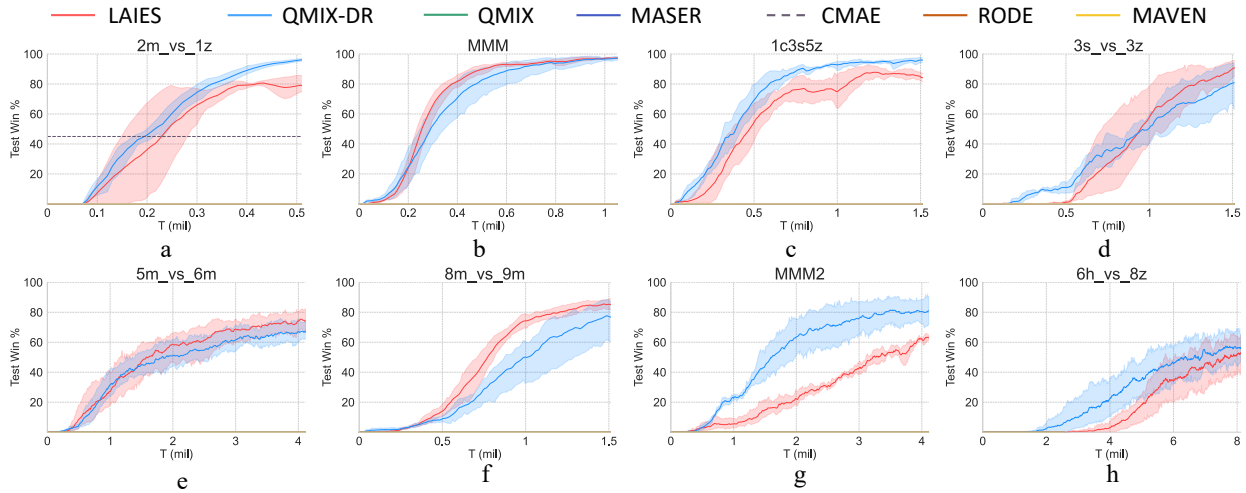


Figure 4. Comparison of our method against baseline methods on eight maps in StarCraft II with the evaluation index of test winning rate.

the enemy has 6 Marines. The comparison of the learning curves in Figure 4.e clearly illustrates the superior performance of our method over QMIX-DR in terms of both convergence and final performance, achieving nearly ten percentage points higher than QMIX-DR.

6h_vs_8z is a super hard map, the ally has 6 Hydralisks, and the enemy has 8 Zealots. LAIES still shows superior performance in this task with acceptable variance and a final winning rate.

As depicted in Figure 4, although the QMIX, RODE, and MAVEN methods exhibit exceptional performance in dense reward scenarios in SMAC, they consistently underperform in all sparse reward scenarios. Although MASER achieves good performance when the reward is not dense by generating subgoals from the experience replay buffer, it fails to perform effectively when the reward is highly sparse. The proposed method LAIES demonstrates superior performance on all tasks, including easy, hard, and super hard maps. In SMAC, a complex reward function (given in in Appendix A.2) is built using expert knowledge, primarily considering the number of defeated enemies and their health value. QMIX-DR utilizes this reward function for learning. While QMIX can learn effective strategies in scenarios with dense rewards, it still performs worse than LAIES in four scenarios, i.e., MMM, 3s_vs_3z, 5m_vs_6m and 8m_vs_9m. These four scenarios share a common characteristic: a relatively small gap in the strength of the enemy and our forces. For example, in MMM, the strength of both sides is symmetrical. By comparing Figure 4.e and Figure 4.f, we can see that the initial gap in strength between the enemy and our forces decreases, and the performance advantage of LAIES over QMIX-DR also increases. We know that laziness is more likely to arise when one team has a significant advantage over the other in real-world teamwork. In SMAC, the reward shaping is feature-based and does not attribute damage to specific agents, which provides an opportunity for lazy behavior. While LAIES's intrinsic motivation is to consider the causal impact of each agent's behavior on the next external state, it encourages each agent to participate in the team cooperation and participate in changing external states. Therefore, in tasks where lazy agents are more likely to occur, LAIES can achieve better results, which also reflects the fact that LAIES can indeed avoid the generation of lazy agents.

5.3. Performance on Google Research Football

In this section, we evaluate our approach on four GRF tasks. Compared to SMAC, GRF emphasizes team collaboration, where agents must learn collaborative skills such as off-ball moving and passing to score. As shown in Figure 5, LAIES surpasses other methods in terms of convergence speed and final win rate across all tasks.

As shown in Figure 5(a), LAIES exhibited the fastest convergence and highest final win rate among the compared methods. QMIX closely followed LAIES in performance, while CDS and QPLEX demonstrated effective strategies. MAVEN requires more time to explore practical strategies. Notably, LAIES displayed a lower variance across random seeds and achieved convergence earlier than QMIX.

In the academy 3vs2 task, the agents face two defenders in front of the goal and must coordinate their passing and movement to bypass the defense. Due to the difficulty level being set to 1, our ball-holding agents are subject to active interception attempts from the opponents. As shown in Figure 5(b), compared to the other four methods, LAIES consistently performs in this task, ultimately achieving a win rate above 40%.

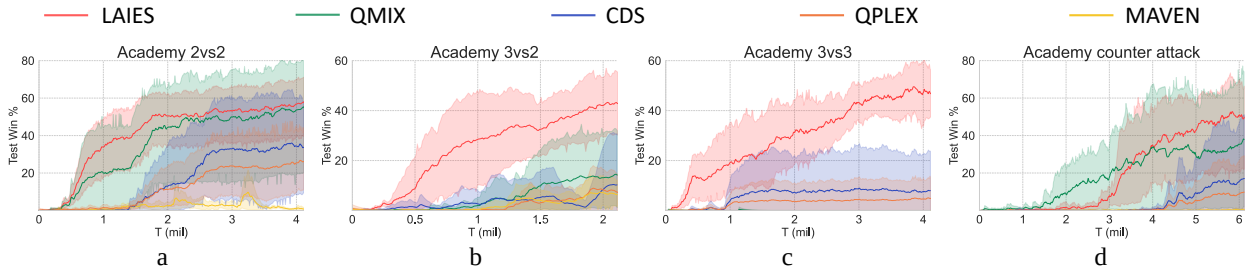


Figure 5. Comparison of our method against baseline methods on four tasks in GRF with the evaluation index of test winning rate.

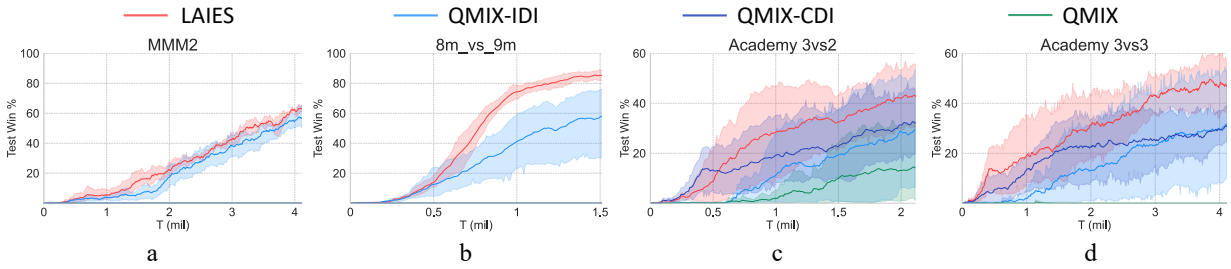


Figure 6. Ablation studies on two maps in SMAC and two tasks in GRF with the evaluation index of test winning rate.

In the academy 3vs3 task, three defenders in front of the goal make it challenging for the agents to score directly. As illustrated in Figure 5(c), QMIX and MAVEN cannot discover effective scoring strategies in this scenario. CDS and QPLEX discover effective strategies in several random seeds but are limited to local optima. However, LAIES still learns effective strategies in this task.

In the academy counterattack task, the agents control four players on the right half of the field and face two defenders and a goalkeeper on the right half of the field. As demonstrated in Figure 5(d), LAIES still outperforms other methods. Despite a slower exploration of scoring strategies compared to QMIX, LAIES can converge to a higher win rate at a faster rising.

5.4. Ablation Studies

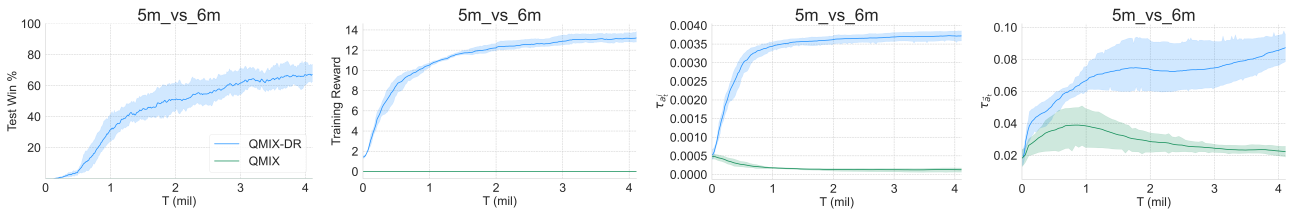
As shown in Figure 6, we conducted an ablation study to evaluate the individual contributions of IDI and CDIt. QMIX-IDI represents QMIX with IDI. QMIX-CDI means QMIX with CDI. We choose four representative tasks to conduct ablation studies.

MMM2 and 8m_vs_9m are tasks in SMAC. Here, the emphasis is on each agent's impact on opponents, and each agent needs to work together to damage the enemy. Lazy agents may cause task failure and collective punishment. As shown in Figure 6 (a) and 6 (b), the removal of IDI significantly drops performance on MMM2 and 8m_vs_9m. These results indicate that IDI is crucial in enabling the agent to learn effective strategies in SMAC. Furthermore, the comparison of LAIES and QMIX-IDI means that CDI allows LAIES to learn and converge to better strategies. This is consistent with our proposed objective of CDI, which aims to improve training by encouraging agents to collectively select joint actions that have a more significant causal effect on external states.

Academy 3vs2 and academy 3vs3 are tasks in GRF. As shown in Figures 6(c) and 6(d), the absence of the IDI or CDI component significantly decreases performance, indicating that both components play a crucial role in final learning results. The comparison between the QMIX-CDI and QMIX-IDI curves highlights that CDI plays a more dominant role in GRF tasks. The QMIX-CDI and LAIES begin to learn effective strategies after similar training steps and earlier than QMIX-IDI. This is in contrast to the results obtained in the SMAC, which may be because the latter environment emphasizes the individual's impact on external states, as success in this environment requires each agent to attack the enemy. On the other hand, in GRF tasks, the changes in external states are holistic, such as the opponent considering all our players before making a decision. Therefore, in GRF, it is more critical for agents to influence external states jointly.

5.5. The Relationship of External States and Diligence

In this study, we suppose that agents with higher levels of diligence will have a stronger influence on external states. We experimented with the 5m_vs_6m task in SMAC to vali-



(a) Test Win Rate (b) Training Rewards (c) IDD (d) CDD

Figure 7. Experimental results on 5m_vs_6m task in SMAC. Figures (c) and (d) are the change curve of IDD and CDD of QMIX and QMIX-DR during training.

date this proposition.

As shown in Figure 7(a), QMIX cannot learn practical strategies unless dense rewards are added (QMIX-DR). In Figure 7(c), the IDD curve of QMIX continually decreases as the training progresses and eventually converges to a value close to 0. Figure 7(b) shows that its rewards remain at 0; whereas in QMIX-DR, the IDD curve continuously increases, in line with the upward trend of the reward curve in Figure 7(b). The experimental results in Figure 7(c) show that the IDD remains lower when the agent cannot complete the task. In contrast, when the agent can complete the task, the IDD remains at a higher level, and as the win rate improves, the IDD also increases. These experimental results demonstrate that the agent's behavior significantly impacts external states as the strategy improves. As shown in Figure 7(d), the trend of the curve is similar to that in Figure 7(c), and the results also indicate that failed strategies have a smaller impact on external states. Still, as the strategy's win rate increases, its effects on external states will also increase. Additionally, in Figures 7(c) and 7(d), the QMIX-DR curves are consistently higher than the QMIX curves, which is consistent with the results in Figures 7(a) and 7(b). This suggests that our metric effectively reflects the diligence of the agents and the team.

In conclusion, these experimental findings show that diligent agents have a greater influence on external states and show that IDD and CDD can accurately reflect the agents' diligence.

6. Conclusions & Future Work

This paper investigates the sparse reward problem in MARL with a new perspective, i.e., lazy agents. Taking the 3m task in SMAC as an example, we analyze why QMIX fails in sparse reward scenarios. Utilizing an agent-environment interaction causal graph, we identified the causes of lazy agents and provided a mathematical definition for them in the context of MARL. To address this issue, we proposed two intrinsic rewards, IDI and CDI, which encourage agents and teams, respectively, to exert influence on external states. Through experiments, we demonstrate the effectiveness of our proposed method compared to SOTA methods.

This paper has two main limitations. Firstly, it is possible to categorize states as internal and external. Most existing MARL scenarios are team competition scenarios, where agents must cooperate to defeat opponents. The external states can be selected as the opponent states in these scenarios. However, in those tasks without opponents, the choice of external states may require domain knowledge. Secondly, this paper defines 'lazy agents' regarding their influence on external states, but their laziness may not be limited to not impacting external states. For example, for the role of Medivac in MMM tasks of SMAC, its laziness is reflected in not impacting the states of alliance agents. Therefore, both the definition and solution of lazy agents are worthy of further studies.

Acknowledgements

This work was supported by the National Key Research and Development Program of China under Grant 2020AAA0103404, the Beijing Nova Program under-Grant 20220484077, the External Cooperation Key Project of Chinese Academy Sciences under Grant 173211KYSB20200002, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27030204.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Bolander, T., Engesser, T., Mattmüller, R., and Nebel, B. Better eager than lazy? how agent types impact the successfulness of implicit coordination. In *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2018.

Connerney, J., Espley, J., Lawton, P., Murphy, S., Odom, J., Oliversen, R., and Sheppard, D. The maven magnetic

field investigation. *Space Science Reviews*, 195(1):257–291, 2015.

Foerster, J. N., Assael, Y. M., De Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

Fuji, T., Ito, K., Matsumoto, K., and Yano, K. Deep multi-agent reinforcement learning using dnn-weight evolution to optimize supply chain performance. 2018.

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

Jeon, J., Kim, W., Jung, W., and Sung, Y. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International Conference on Machine Learning*, pp. 10041–10052. PMLR, 2022.

Kim, D.-K., Riemer, M., Liu, M., Foerster, J., Everett, M., Sun, C., Tesauro, G., and How, J. P. Influencing long-term behavior in multiagent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:18808–18821, 2022.

Kurach, K., Raichuk, A., Stańczyk, P., Zajac, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., et al. Google research football: A novel reinforcement learning environment. *arXiv preprint arXiv:1907.11180*, 2019.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Letcher, A., Foerster, J., Balduzzi, D., Rocktäschel, T., and Whiteson, S. Stable opponent shaping in differentiable games. *arXiv preprint arXiv:1811.08469*, 2018.

Li, C., Wu, C., Wang, T., Yang, J., Zhao, Q., and Zhang, C. Celebrating diversity in shared multi-agent reinforcement learning. *arXiv preprint arXiv:2106.02195*, 2021.

Li, P., Tang, H., Yang, T., Hao, X., Sang, T., Zheng, Y., Hao, J., Taylor, M. E., and Wang, Z. Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration. *arXiv preprint arXiv:2203.08553*, 2022.

Li, S. Multi-agent deep deterministic policy gradient for traffic signal control on urban road network. In *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pp. 896–900. IEEE, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Liu, I.-J., Jain, U., Yeh, R. A., and Schwing, A. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6826–6836. PMLR, 2021.

Liu, M., Zhou, M., Zhang, W., Zhuang, Y., Wang, J., Liu, W., and Yu, Y. Multi-agent interactions modeling with correlated policies. *arXiv preprint arXiv:2001.03415*, 2020.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.

Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.

Oliehoek, F. A. and Amato, C. *A concise introduction to decentralized POMDPs*. Springer, 2016.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Perrusquía, A., Yu, W., and Li, X. Multi-agent reinforcement learning for redundant robot control in task-space. *International Journal of Machine Learning and Cybernetics*, 12(1):231–241, 2021.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10199–10210, 2020.

Rosenbaum, C., Klinger, T., and Riemer, M. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *arXiv preprint arXiv:1711.01239*, 2017.

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

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019b.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tang, H., Hao, J., Lv, T., Chen, Y., Zhang, Z., Jia, H., Ren, C., Zheng, Y., Meng, Z., Fan, C., et al. Hierarchical deep multiagent reinforcement learning with temporal abstraction. *arXiv preprint arXiv:1809.09332*, 2018.

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

Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., and Zhang, C. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020b.

Xie, A., Losey, D., Tolsma, R., Finn, C., and Sadigh, D. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pp. 575–588. PMLR, 2021.

Yang, Y., Hao, J., Liao, B., Shao, K., Chen, G., Liu, W., and Tang, H. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

A. Experiment Details

A.1. LAIES and Baselines

In this paper, we compare our approach with several methods. LAIES is developed based on the QMIX. For QMIX and QPLEX, we use the code framework in https://github.com/hijkzzz/pymarl2. LAIES is also implemented based on this code framework. Except for the additional parameters in LAIES, all other parameters are set the same with QMIX, such as batch size, learning rate, parallel environments, etc. For MAVEN, we use the code in https://github.com/starry-sky6688/MARL-Algorithms. For CDS, MASER and RODE, we use the code provided by the author. For SMAC and GRF tasks, we ensure the same environmental settings as LAIES, including reward, observation, states settings, avail actions, etc. It is noted that LAIES, QMIX and QPLEX using 8 parallel environments (parallel runner) whereas RODE, MAVEN and CDS using 1 parallel environment (episode runner).

In LAIES, we introduce two important hyperparameters: β_1 and β_2, correlated to the MI regularizers. For SMAC scenarios, we use $\{\beta_1,\beta_2\} = \{100, 0.2\}$ on 3m, 1c3s5z, 3s_vs_3z, 8m_vs_9m, MMM2, and 6h_vs_8z, $\{\beta_1,\beta_2\} = \{600, 0.3\}$ on 2m_vs_1z, $\{\beta_1,\beta_2\} = \{200, 0.3\}$ on 5m_vs_6m, $\{\beta_1,\beta_2\} = \{20, 0.02\}$ on MMM. For GRF scenarios, we also search the best hyperparameters, and use $\{\beta_1,\beta_2\}$=$\{1,8\}$ on all tasks. It is noted that intrinsic rewards are annealed after mean rewards over 0.

Experiments are carried out on NVIDIA GTX3090 GPU.

A.2. SMAC

StarCraft II (Samvelyan et al., 2019b) is a popular real-time strategy game, which derives many micromanagement scenarios. In the micromanagement scenarios, the agents need to cooperate to eliminate the enemies. This benchmark consists of various maps classified as easy, hard, and super hard. We test our method on nine micromanagement tasks i.e., 3m, 2m_vs_1z, MMM, 1c3s5z, 3s_vs_3z, 5m_vs_6m, 8m_vs_9m, MMM2, and 6h_vs_8z. Details of these maps are shown in Table 1.

Table 1. SMAC challenges.

Task	Ally Units	Enemy Units	Type	Difficulty
3m	3 Marines	3 Marines	homogeneous, symmetric	easy
2m_vs_1z	2 Marines	1 Zealot	micro-trick: alternating fire	easy
MMM	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 2 Marauders , 7 Marines	heterogeneous, symmetric	easy
1c3s5z	1 Colossi , 3 Stalkers, 5 Zealots	1 Colossi , 3 Stalkers, 5 Zealots	heterogeneous, symmetric	easy
3s_vs_3z	3 Stalkers	3 Zealots	micro-trick: kiting	hard
5m_vs_6m	5 Marines	6 Marines	homogeneous, asymmetric	hard
8m_vs_9m	8 Marines	9 Marines	homogeneous, asymmetric	hard
MMM2	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 2 Marauders , 8 Marines	Asymmetric, Heterogeneous	super hard
6h_vs_8z	6 Hydralisks	8 Zealots	micro-trick: focus fire	super hard

Except for QMIX-DR, we use sparse reward settings in SMAC. All enemies die resulting in a positive reward of +1. Conversely, if all of our allies are eliminated, we receive a negative reward of -1. In instances where both teams have surviving bots at the end of an episode, a neutral reward of 0 is obtained.

In QMIX-DR, there is a a shaped reward based on the hit-point damage on the enemies and a special incentive for winning the battle (SMAC default dense reward setting). The detailed reward of each scenario is defined as follows:

$$r_t = \frac{\sum_{k=1}^{N} \Delta h_k^t + N_{\text{death}}^t \times r_{\text{kill}}}{\sum_{k=1}^{N} H_{\text{total}}^k + N \times r_{\text{kill}} + r_{\text{win}}} \tag{12}$$

where N represents the number of enemies. N_{death}^t represents the number of enemies died at step t. H_{total}^k total is the total

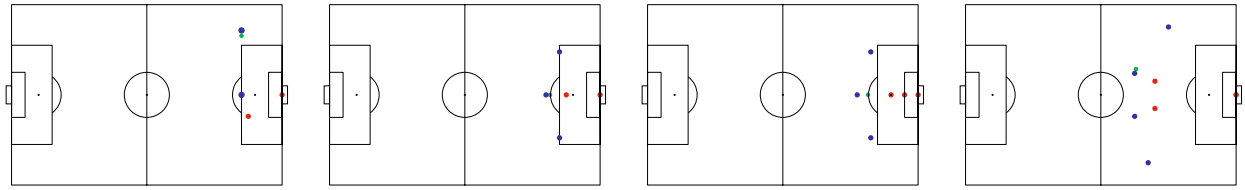
health of enemy k. $\Delta h_k^t = h_k^t - h_k^{t-1}$ is the health difference of enemy k between two steps. r_{kill} and r_{win} are the special bonuses for killing the enemy and winning the battle, which are set as 10 and 200, respectively.

A.3. GRF Tasks

In GRF, agents are trained to play football in a physics-based 3D simulator. GRF is a challenging task for its inner stochasticity and sparse reward. The agents must learn high-level cooperation skills such as passing, obstructing opponents for teammates, et al., and then score a goal. We choose four academy tasks (3 official and 1 hand-crafted) to evaluate our method, i.e., academy run pass and shoot with keeper (abbreviated as academy 2vs2), academy 3vs1 with keeper (abbreviated as academy 3vs2), academy counterattack hard (abbreviated as academy counterattack), and academy 3v3. In all these tasks, the offside is prohibited and the difficulty is set as 1 in academy 2vs2 , academy 3vs2 and academy 3vs3. In these tasks, external states refer to opponents' and ball's positions and directions.

The initial positions of players, opponents, and the ball are shown in Fig. 8. In these tasks, we control the left team, where each agent must choose an action from available actions, including run, pass, dribble, shot, etc. We have rewritten the available actions based on expert knowledge. For example, shooting is allowed only when the player is close to the goal and holding the ball. All agents must cooperate well to organize offenses and seize fleeting opportunities. There are only two types of rewards: (1) a reward +10 for the left team to score a goal; (2) a reward -5 for the left team failing to score a goal. An episode will be terminated when reaching the following four situations: (1) the ball controlled by opponents, (2) the ball returning to left half-court, (3) scoring a goal (4) the ball bouncing out of fields. The observation contains the positions and directions of the ego-agent, teammates, and the ball.

During training, we ensure that LAIES and other algorithms, including QMIX, CDS, QPLEX, and MAVEN, have the same environment settings, including the available actions, reward settings, observations, and states.



(a) academy 2vs2 (b) academy 3vs2 (c) academy 3v3 (d) academy counterattack

Figure 8. Visualization of the initial position of each agent in five GRF tasks. Blue dots represent the agent. Red dots are opponents, and the green dot denotes the ball.

B. More Experimental Results

B.1. LAIES with IPPO

In the previous experimental analysis, the combination of LAIES and QMIX, a value-based method, significantly improves its training performance in sparse reward scenarios. To further verify the effectiveness of LAIES in policy gradient methods, we combine LAIES with IPPO and test its performance in four SMAC tasks, i.e., 3m, 2m_vs_1z, MMM and 1c3s5z. To ensure a fair comparison, all parameters are kept consistent with IPPO, except for the additional parameters introduced by LAIES.

As shown in Figure 9, in the 2m_vs_1z and 3m tasks, LAIES achieved better results than any other method. In the MMM task, LAIES outperforms IPPO-DR in terms of variance and final win rate and learns strategies to defeat opponents earlier. In the 1c3s5z task, LAIES performs far better than IPPO-DR, learning a strategy to kill all enemies with a probability of nearly 80% after one million training steps. In contrast, IPPO-DR can only learn a strategy to defeat enemies with a probability of nearly 20%.

Overall, the experimental results demonstrate that LAIES can effectively improve the training of policy gradient algorithms in sparse reward scenarios and achieve outstanding results in StarCraft tasks.

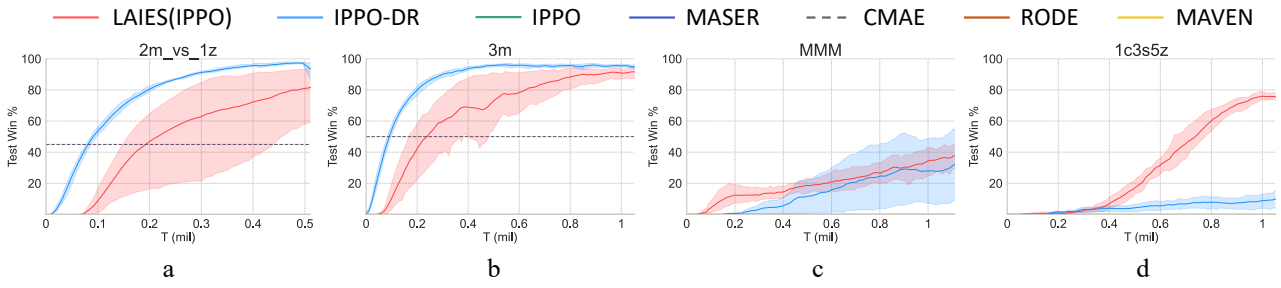


Figure 9. Comparison of our method against baseline methods on four tasks in SMAC with the evaluation index of test winning rate.