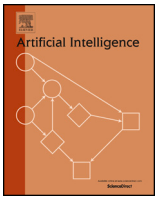
Artificial Intelligence 346 (2025) 104357

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Active legibility in multiagent reinforcement learning

Yanyu Liu [a], Yinghui Pan [b,*], Yifeng Zeng [c,*], Biyang Ma [d], Prashant Doshi [e]

[a] School of Automation, Central South University, No. 605 South Lushan Road, Changsha, 410083, Hunan, China
[b] School of Artificial Intelligence & National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, Guangdong, China
[c] Department of Computer and Information Sciences, Northumbria University, Newcastle, United Kingdom
[d] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, China
[e] Department of Computer Science, University of Georgia, Athens, GA, USA

ARTICLE INFO

Keywords:
Legibility
Multiagent reinforcement learning
Multiagent interaction

ABSTRACT

A multiagent sequential decision problem has been seen in many critical applications including urban transportation, autonomous driving cars, military operations, etc. Its widely known solution, namely multiagent reinforcement learning, has evolved tremendously in recent years. Among them, the solution paradigm of modeling other agents attracts our interest, which is different from traditional value decomposition or communication mechanisms. It enables agents to understand and anticipate others' behaviors and facilitates their collaboration. Inspired by recent research on the legibility that allows agents to reveal their intentions through their behavior, we propose a *multiagent active legibility framework* to improve their performance. The legibility-oriented framework drives agents to conduct legible actions so as to help others optimize their behaviors. In addition, we design a series of problem domains that emulate a common legibility-needed scenario and effectively characterize the legibility in multiagent reinforcement learning. The experimental results demonstrate that the new framework is more efficient and requires less training time compared to several multiagent reinforcement learning algorithms.

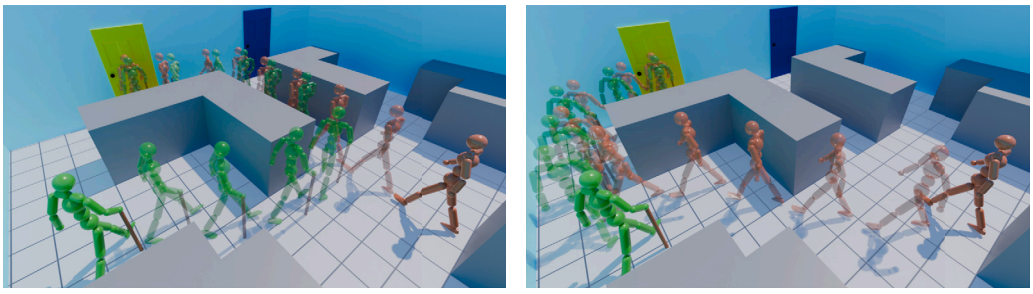
1. Introduction

Multiagent Reinforcement Learning (MARL), as a powerful method to tackle multiagent sequential decision problems, has grown tremendously in the past two decades [1]. As MARL research develops, cooperative tasks have emerged as one of the primary focuses. Enabling agents to learn cooperative behaviors facilitates the completion of more complex tasks, thereby providing more benefit to human life. When MARL evolves to learn agents' collaboration, it often leads to two branches: value decomposition and centralized-critic. The value decomposition methods train a global Q-network with the consideration of global information and are able to overcome the MARL instability, e.g. Value-Decomposition Networks (VDN) [2], QTRAN [3], QMIX [4]. The centralized-critic methods aim to learn a centralized critic network to train a distributed actor policy, e.g. Mutilagent Deep Deterministic Policy Gradient (MADDPG) [5], Counterfactual Multiagent Policy Gradients (COMA) [6].

* Corresponding authors.
E-mail addresses: liuyy99.cn@csu.edu.cn (Y. Liu), panyinghui@szu.edu.cn (Y. Pan), yifeng.zeng@northumbria.ac.uk (Y. Zeng), mby@mnnu.edu.cn (B. Ma), pdoshi@uga.edu (P. Doshi).

https://doi.org/10.1016/j.artint.2025.104357
Received 23 October 2024; Received in revised form 1 May 2025; Accepted 2 May 2025

Available online 19 May 2025



(a) Illegible trajectory (b) Legible trajectory

Fig. 1. An example of utilizing legibility in a multiagent system: different paths taken by red humanoid lead to different goal predictions by green humanoid, resulting in distinct collaboration outcomes. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Beyond those approaches, another promising direction in MARL is modeling intentions and policies of other agents. By understanding and anticipating teammates' behaviors, agents can better coordinate their actions and optimize collective performance. Recently, several frameworks [7–9] have emerged, focusing on modeling other agents. Wen et al. [10] presented a probabilistic recursive reasoning (PR2) framework, which adopted variational Bayes methods to approximate the opponents' future actions, where every agent found the best response and then improved its own policy. Empirically, predicting the intended end-product (goal), as well as the sequence of steps (plan), can be helpful for improving the performance of agents' interaction in MARL. Yu et al. [11] mentioned that predicting actions is shortsighted and limits generalization to unknown opponents. They proposed the opponent modeling based on sub-goal inference (OMG) framework, which utilizes Bayesian inference over historical trajectories to predict opponents' sub-goals, achieving effective adaptation. The premise is that the system's overall objective can be decomposed, with individual agents to achieve their sub-goals. Various techniques have been proposed for task allocation among agents in recent years. For instance, Tian et al. [12] proposed the framework of decomposing a task into a series of generalizable subtasks (DT2GS), which decomposes the source task into independent subtasks such as "hit and run," "focus fire," etc.

By recognizing others' plans, agents can adapt their own behaviors to support or complement their teammates' goals therefore leading to more efficient and effective collaboration. However, solely relying on the agents' modeling capability has limitations and faces numerous challenges in enhancing system performance. One of the challenges is *ambiguity*: extracting the plan or predicting the future actions from observed trajectories could be confusing and complex, as trajectories may have multiple interpretations or could be misinterpreted by unforeseen factors. Once the agent recognizes the incorrect intention, the consequence could be devastating and catastrophic. On the other hand, the contradiction between the recognition accuracy and computational complexity severely limits the generalization of plan recognition in MARL.

Recently, *legibility* has been introduced to facilitate agents to convey their intentions through their behaviors [13]. The legibility could reduce the ambiguity over the possible goals of agents from an observer's perspective and improve human-and-agent collaboration. The most recent work [14], namely Information Gain Legibility (IGL), used the information entropy gain to shape the reward signal and improved the *agent-to-human* policy legibility. Compared to the existing methods [15,16], IGL is much easier to implement and can theoretically be extended to many reinforcement learning methods. However, in multiagent settings, reducing information entropy does not guarantee that the goal which an observer believes most is aligned with the agent's true goal (i.e., the observer could strongly believe in an incorrect goal). Thus, we will investigate the MARL legibility further in this paper.

We begin with an example to show how the legibility facilitates multiagent collaboration. The example of Fig. 1 includes red/green humanoid agents, uncrossable obstacles, and doors that require both the agents to open together. The red agent knows the target door at the beginning. In contrast, the green agent on crutches moves slower and is unaware of the target door. Besides, those agents can only observe each other's positions and orientations, without additional information such as gestures or voice. This creates a *leader-follower* dynamic, where the leader (red agent) guides the follower (green agent) to the target. Since the follower is slower, following the leader **step-by-step** is insufficient to achieve the shortest completion time. The slow agent can infer the target door by analyzing the fast agent's trajectory and then choose a shorter path to the potential target [17,18]. In this way, the green agent can arrive in advance and wait for the red agent, thus reducing overall task completion time and steps.

Fig. 1 illustrates the different trajectories of tasks completed by two agents under the same initial conditions. In Fig. 1(a), the red agent first moves straight through the corridor and then turns left towards the door. During this time, the green agent is unable to determine whether the red agent's target is the yellow or blue door until the red agent exits the corridor. As a result, the green agent must either follow the red agent step by step or remain stationary until the target is identified, before taking any further action. In contrast, in Fig. 1(b), the red agent first moves forward and to the left, then turns right towards the yellow door. In this case, by improving the legibility of the action sequence, the green agent can eliminate erroneous targets and infer the true target shortly after game begins, thereby reducing overall time and distance.

The example shows that for some specialized multiagent tasks, introducing the legibility can reduce the ambiguity in agents' modeling, providing advantages that cannot be achieved solely by improving agents' learning capabilities. This raises the upper limit of task completion performance. Furthermore, the legibility can also increase the RL policy transparency and interpretability, which is highly significant for safety-critical domains such as military defense, health, finance, etc.

Driven by the aforementioned concepts, we contribute a novel approach to MARL by conducting the first exploration on the MARL legibility. In this paper, we present the Multiagent Active Legibility (MAAL) framework that exploits the legibility in MARL from individual agents' perspectives. In MAAL, a subject agent adapts a reward shaping technique to conduct legible actions by narrowing down the margin between the predicted actions from other agents and the true ones to be executed by the subject agent. Compared to the previous IGL technique, MAAL improves the *agent-to-observer* legibility in a RL agent and seeks to achieve seamless cooperation between the agent and observer. Furthermore, unlike IGL which leverages information entropy of the observer's beliefs to shape the reward function, MAAL uses the variation in the Kullback–Leibler (KL) divergence between the observer's beliefs and the true goal before and after executing their actions, and instructs the agent to derive the legibility reward function.

The main contributions of this paper are as follows:

- We propose a Multiagent Active Legibility (MAAL) framework and define a Legible Interactive-POMDP (LI-POMDP), which enhances the legibility of individual agents' actions, thereby improving MARL collaboration.
- We develop a series of problem domains that simulate typical scenarios of legibility, effectively showcasing the importance of legibility in MARL.
- We conduct extensive comparative experiments, ablation study, and theoretical analysis, demonstrating the convergence and effectiveness of the MAAL framework.

The remainder of this paper is organized as follows. Section 2 examines relevant MARL research. Section 3 introduces background knowledge pertaining to multiagent decision-making and the policy legibility. We present the new MAAL framework in Section 4. Section 5 offers a precise completeness analysis of the legibility approach in MAAL. The experimental settings and result analysis are presented in Section 6. Lastly, we summarize this research and brief the future work in Section 7.

2. Related works

In this section, we first present the current mainstream methods in multiagent reinforcement learning in Section 2.1. In Section 2.2, we discuss methods for modeling other agents. Finally, in Section 2.3, we brief the legibility concept and review how it influences agent's behavior and decision-making.

2.1. Multiagent reinforcement learning

Referring to multiagent reinforcement learning, there has seen a series of classical methods. Sunehag et al. [2] proposed the VDN framework, which views the system's overall value as the summation of the individual value of agents. VDN substantially overcomes the non-stationarity of MARL in a concise way. However, the linear summation may not accurately represent the overall value in some complex environments. QMIX [4] inherits the ideology and approximates a monotonic function between a local Q-network and an overall Q-network with a nonlinear neural network. Son et al. [3] proposed QTRAN to decompose a joint Q-value function into a sum of task-specific value functions and used task-relational matrices to transfer knowledge between the tasks - being free from the subjective *Additivity* and *Monotonicity*.

In parallel, a lot of research expands the policy gradient method to MARL applications. Lowe et al. [5] proposed MADDPG method, where agents learned to estimate their own action-value function by utilizing a centralized critic network that took into account the joint actions of all the agents. Foerster et al. [6] introduced COMA method that used the counterfactual baseline, which attempted to estimate what the value would have been with alternative joint actions. This type of research has achieved promising results in solving complex MARL problems such as StarCraft Multiagent Challenge.

2.2. Modeling other agents

In human cooperation activities, persons subconsciously predict others' next moves, or intentions to rectify their own policy. Gmytrasiewicz et al. [19] proposed interactive POMDP (I-POMDP) that introduced models of other agents into the partially observed MDP. In I-POMDP, an agent is able to construct belief models regarding its understanding of the knowledge and beliefs held by other agents. However, solving the I-POMDP is rather difficult due to the inherent complexity even with purpose-designed methods [20–22] including its graphical counterpart - interactive dynamic influence diagrams [23].

Meanwhile, several opponent modeling techniques have been developed. Yang et al. [8] introduce Bayesian Theory of Mind on Policy (Bayes-ToMoP) that efficiently detected opponents' non-stationary and sophisticated strategies by employing Bayesian Policy Reuse [24] and Theory of Mind [25]. Tian et al. [9] presented an opponent modeling algorithm with a novel objective, namely Regularized Opponent Model with Maximum Entropy Objective (ROMMEO), which extended a Maximum Entropy Objective to multiagent settings and frames MARL as Bayesian inference problems. Wen et al. [10] proposed Probabilistic Recursive Reasoning (PR2) that implemented the recursive reasoning in MDP, and encouraged the modeling method from game theory [26,27]. Later, they proposed Generalized Recursive Reasoning (GR2) [28] which took bounded rationality into the consideration and extended the reasoning level to an arbitrary number. However, the inherent computational and action complexity of RR limits its application and, in some scenarios, the recursive reasoning can even be NP-hard. On the contrary, plan recognition is not only more computation-friendly, but less reliant on a complete and accurate model of beliefs, preferences, and actions, as well as mutual beliefs and knowledge among other agents [29].

2.3. The legibility for intelligent agents

Most of the existing research on the legibility arises from intention recognition in the motion control in robotics while the recent work develops a transparent, rigorous and distinct representation of memory mechanism to drive observable actions in brain neural networks [30]. Dragan et al. [16] presented the mathematical definition of legibility, and proposed a model in which the agent could evaluate or generate motion by the functional gradient optimization in the space trajectory. Holladay et al. [31] studied the implication of legibility in robot pointing, e.g. producing pointing gestures. Nikolaidis et al. [32] explored the impact of the observer's viewpoint on the legibility of motion and trajectories and proposed a viewpoint-based strategy for optimizing legibility. They offered new insights into using occlusion to generate deliberately ambiguous or deceptive movements. Bied et al. [33] proposed a reward-shaping method that embedded an observer model within reinforcement learning to enhance the legibility of an agent's trajectory. Busch et al. [34] introduced a model-free reinforcement learning algorithm to optimize a general, task-independent cost function, together with an evaluation strategy to determine whether the learned behavior is universally legible.

Recently, the legibility research has been extended to agents' policy learning and decision-making under uncertainty. Persiani et al. [35] proposed a method that utilizes Bayesian Networks to model both the agent and the observer. They optimized the legibility of the agent model by minimizing the cross-entropy between these two networks. Bernardini et al. [36] presented a joint design for goal legibility and recognition in a cooperative, multiagent scenario with partial observability. Zhao et al. [37] conducted legibility tests within the context of Deep Reinforcement Learning (DRL) and suggested the use of Recurrent Neural Networks (RNN) as motion predictors to score the legibility of actions. Miura et al. [38,13] extended the work on maximizing legibility from deterministic settings to stochastic environments, and introduced a Legible Markov decision problem (L-MDP). They also proposed Observer-Aware MDP (OAMDP) [39] as a less complex framework of I-POMDP under certain conditions and extended it with explicit communication as Communicative Observer-Aware MDP (Com-OAMDP) [40]. Faria et al. [41] proposed a more computation-friendly framework compared to L-MDP, namely Policy Legible MDP (PoLMDP), which considerably lowered the complexity by solving nearly a standard MDP.

3. Preliminary

In this section, we introduced the preliminary knowledge related to MAAL, starting with the fundamental Markov Decision Process, followed by two related Partial Objective Markov Decision Processes: the I-POMDP, which introduces agent modeling, and the PoLMDP, which incorporates legibility in agents.

3.1. Markov decision process and I-POMDP

The problem of learning a goal-directed policy is typically formulated into a mathematical framework known as Markov Decision Process (MDP), which is a discrete-time stochastic process with the Markovian property. In MDP, the future state depends solely on the current state and the action taken, independent of past states and actions. In this paper, we define MDP as a tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma\}$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{T}$ denotes transition probability, R denotes the reward signal, and $\gamma \in [0, 1]$ denotes the discount factor. In traditional single-agent RL, we seek to find a policy π that maximizes the expected return over the states $s \in \mathcal{S}$.

$$\mathcal{J}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t] \tag{1}$$

In MARL, the environment is non-stationary from the perspective of each agent because other agents are learning and changing their policies simultaneously. This makes it challenging for the agents to learn stable policies. From individual agents' perspectives, interactive-POMDP enables agents to utilize more sophisticated techniques to model and predict behaviors of other agents [19]. An I-POMDP, for the subject agent $A^i \in \{A^1, ..., A^N\}$, is defined as $\{IS_i, \mathcal{A}, \mathcal{T}, \Omega_i, \mathcal{O}_i, R_i\}$, in which the most notable component is the interactive state $IS_i = \mathcal{S} \times M$, where M holds all possible models of other agents. Apart from this, other components of I-POMDP are similar to a standard POMDP. The main feature of I-POMDP lies in incorporating the modeling of other agents into the subject agent's decision optimization, enabling the agents to observe and coordinate with each other.

3.2. Policy legible Markov decision process

Recently, Faria et al. [41] introduced legibility into traditional MDPs, namely Policy Legible Markov Decision Process (PoLMDP), denoted as:

$$PolMDP = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, R, \gamma, \theta\} \tag{2}$$

A PoLMDP is defined in the context of an environment with N different objectives, each of which is represented by a different reward function $R_n (n = 1, \cdots, N)$ and thus with a different MDP $\mathcal{M}_n$. $\mathcal{R}$ is the legible reward function, denoted as:

$$\mathcal{R}(s, a) = P(R_n | s, a) \tag{3}$$

where $P(R_n|s, a)$ can be reformulated via Bayes' Theorem below.

$$P(R_n|s,a) \propto P(s,a|R_n)P(R_n) \tag{4}$$

This transformation allows to express the conditional probability of the reward R_n given the state s and action a in terms of the conditional probability of the state and action given the reward, multiplied by the prior probability of the reward. $P(s,a|R_n)$ can be determined by applying the maximum-entropy principle adopted by Dragan [15], with θ serving as the hyper-parameter controls how close the legible function follows the optimal expected reward. Here, Q_n^* represents the optimal Q-function for the MDP $\mathcal{M}_n$, and is calculated as follows:

$$P(s,a|R_n) = \frac{\exp(\theta Q_n^*(s,a))}{\sum_{m=1}^{N} \exp(\theta Q_m^*(s,a))} \tag{5}$$

PoLMDP has the advantage on the computational simplicity therefore easing the implementation. However, PoLMDP requires the optimal policy under the current objective during the training, which is challenging in multiagent environments where each agent's policy is constantly updated. Moreover, the completeness of PoLMDP is difficult to be guaranteed. Hence, by expanding its framework, we propose a novel reward shaping function that is better suited for multiagent interaction. Additionally, we conduct a formal analysis of convergence and completeness on achieving the legibility.

4. Active legibility through reward shaping

Single-agent reinforcement learning has been extensively studied; however, the dynamics become significantly more complex when multiple agents coexist within a shared environment. The most challenging is the non-stationarity in a multiagent setting, which is caused by the changing of other agents' policies over time as they learn. Upon the previous exploration of legibility, we first define the Legible Interactive POMDP (LI-POMDP) in Section 4.1, which provides the theoretical foundation for the subsequent work. In Section 4.2, we propose the Multiagent Active Legibility (MAAL) framework. The MAAL framework is an implementation of LI-POMDP and aims to improve the legibility of individual agents' actions to other agents, thus enabling the subject agent to be more easily modeled by others.

4.1. An interactive POMDP framework with legibility

We consider a set of N collaborative agents $\{A^1, A^2, \ldots, A^N\}$ working together to achieve a common goal g^* in a shared environment. We assume that the goal g^* can be decomposed into N sub-goals, such that $g^* = \bigcup_{i=1}^{N} g^i$, where each sub-goal $g^i \in \mathcal{G}^i$ is assigned to an individual agent $A^i \in \{A^1, A^2, \ldots, A^N\}$. The completion of all sub-goals leads to the successful achievement of the overall task.

To facilitate the development of legibility, we extend the POMDP framework and define the Legible Interactive POMDP (LI-POMDP) as follows. This extension aims to enhance the generation of legible policies by enabling agents to exploit the beliefs of other agents about the subject agent's goals.

Definition 1. For a subject agent $A^i \in \{A^1, ..., A^N\}$, a legible I-POMDP (LI-POMDP) is defined as:

$$LI\text{-}POMDP_i = \left\{S, \mathcal{A}, \mathcal{T}^i, \mathcal{O}^i, \Omega^i, R^i, \mathcal{G}, \mathcal{B}, \mathcal{I}^i, \mathcal{P}^i, \mathcal{R}^i\right\} \tag{6}$$

where:

- S is the state space of the environment.
- $\mathcal{A} = \mathcal{A}^i \times \mathcal{A}^{-i}$ is the joint action space of the subject agent A^i and other agents A^{-i}.
- $\mathcal{T} : S \times \mathcal{A} \times S \to [0,1]$ is the transition function of multiagent environment.
- $\mathcal{O}^i$ is the set of observations that A^i receive from environment.
- $\Omega^i : S \to \mathcal{O}^i$ is the observation function and controls what A^i can receive in state s.
- R^i is the raw reward signal sent from the environment.
- $\mathcal{G} = \mathcal{G}^i \times \mathcal{G}^{-i}$ is the set of goals in a multiagent system, where $\mathcal{G}^i$ represents the set of sub-goals for the subject agent A^i, and $\mathcal{G}^{-i}$ denotes the set of behavioral goals for other agents A^{-i}. This definition allows for different roles among agents in a multiagent system: $\mathcal{G}^i \neq \mathcal{G}^{-i}$. It also accommodates the scenario where agents have an equal footing, with identical sets of behavioral patterns, i.e., $\mathcal{G}^i = \mathcal{G}^{-i}$.
- $\mathcal{B} = \mathcal{B}^{-i} \times \mathcal{B}^i$ is the belief over goals $\mathcal{G}$, in which $\mathcal{B}^{-i} \in \mathbb{R}^{(N-1)|\mathcal{G}|\times 1}$ is the belief space of A^i's beliefs about all other agents in the system, and $\hat{\mathbf{b}}^{-i} \in \mathcal{B}^{-i}$ detailing A^i's prediction about the goals of A^{-i} *(what A^i thinks about the collective goals of other agents)*. $\mathcal{B}^i \in \mathbb{R}^{|\mathcal{G}|\times 1}$ is the belief space of A^i's prediction of how A^{-i} views A^i ($\hat{\mathbf{b}}^i \in \mathcal{B}^i$). In other words, $\hat{\mathbf{b}}^i$ can be interpreted as *what A^i thinks A^{-i} thinks about A^i*.
- $\mathcal{I}^i : \mathcal{O}^i \times \mathcal{A}^{-i} \longrightarrow \mathcal{B}^{-i}$ is the function for subject agent A^i to infer and predict the goals of other agents A^{-i}.
- $\mathcal{P}^i : \mathcal{O}^i \times \mathcal{A}^{-i} \longrightarrow \mathcal{B}^i$ is the function to indicate how other agents A^{-i} are predicting the subject agent A^i's goal.
- $\mathcal{R}^i$ is the reward shaping function to enhance the legibility of A^i's policy. It is derived from the original reward R^i by the environment (to be elaborated in Section 4.1.2).

We begin with modifying the objective function in Eq. (1) and propose a new legible objective function in Eq. (7).

$$\mathcal{J}(\pi^i) = \mathbb{E}_{\pi^i}[\sum_{t=0}^{\infty} \gamma^t r_t^i - \beta D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_t^i)] \tag{7}$$

where $\mathbf{g}^i \in \mathbb{R}^{|\mathcal{G}|\times 1}$ is the true distribution of agent A^i over the goals $\mathcal{G}$, denoted as $\mathbf{g}^i = f_{\mathbf{1}}(g^i)$, where $f_{\mathbf{1}}$ signifies the one-hot encoding function, $\beta \in \mathbb{R}^+$ is the legibility weight to control the legibility level, $\hat{\mathbf{b}}_t^i \in \mathbb{R}^{|\mathcal{G}|\times 1}$ is the estimation of the predictive distribution of g^i from other agents A^{-i} at the time slice t, and D_{KL} is the Kullback-Leibler divergence between $\hat{\mathbf{b}}_t^i$ and $\mathbf{g}^i$, computed as follows:

$$D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_t^i) = \sum_{g \in \mathcal{G}^i} \mathbf{g}^i(g) \log \frac{\mathbf{g}^i(g)}{\hat{\mathbf{b}}_t^i(g)} \tag{8}$$

Eq. (7) maximizes the discounted cumulative reward and simultaneously seeks to minimize the margin between the other agent's prediction and the true goal of the subject agent.

4.1.1. Modeling and predicting other agents

Plan recognition involves interpreting the intentions and plans of other agents. It largely depends on the analysis of agents' behaviors in context. By harnessing the Bayesian update, we elevate the real-time plan recognition, thereby bolstering the system's overall clarity and transparency.

Initially, we establish the function $\mathcal{I}^i$ for the subject agent A^i to infer the goals of other agents A^{-i} from the observed actions $\mathcal{A}^{-i}$ and observations $\mathcal{O}^i$,

$$\mathcal{I}^i : \mathcal{O}_{t+1}^i \times \mathcal{A}_t^{-i} \longrightarrow \mathcal{B}_{t+1}^{-i} \tag{9}$$

where $\hat{\mathbf{b}}^{-i} \in \mathcal{B}^{-i}$ is the belief of the subject agent A^i over the goals of other agents A^{-i}. The belief $\hat{\mathbf{b}}_t^{-i}$ at time t represents how the subject agent A^i reasons about the underlying intentions of A^{-i} given their historical action sequences $\{a_0^{-i}, \cdots, a_{t-1}^{-i}\}$. If A^{-i} consists of more than one agent, $\hat{\mathbf{b}}^{-i}$ is the concatenated beliefs about the goals of individual agents,

$$\hat{\mathbf{b}}_t^{-i} = \odot_{j\in[1,N]}^{j\neq i} \hat{\mathbf{b}}_t^j \tag{10}$$

where $\hat{\mathbf{b}}_t^j$ represents the belief of A^i regarding the goal of other agent $A^j \in A^{-i}$. In the absence of prior knowledge, we assume that agent A^i holds a uniform belief about the goal of agent A^j, denoted as $\hat{\mathbf{b}}_0^j(g^j) = \frac{1}{|\mathcal{G}|}, \forall g^j \in \mathcal{G}$. Consequently, a function is needed to map a policy to the belief over the agent's goals and update this prediction. To facilitate this process, we adopt the Bayesian approach [42] for the real-time belief update,

$$\begin{aligned} \hat{\mathbf{b}}_t^j(g^j) &= \frac{1}{|\mathcal{G}|}, \forall g^j \in \mathcal{G}, t = 0 \\ \hat{\mathbf{b}}_{t+1}^j(g^j) &= \frac{\hat{\pi}^j(o_{t+1}^i, a_t^j | g^j)}{\sum_{g' \in \mathcal{G}} \hat{\pi}^j(o_{t+1}^i, a_t^j | g') \hat{\mathbf{b}}_t^j(g')} \hat{\mathbf{b}}_t^j(g^j), \forall g^j \in \mathcal{G}, t \neq 0 \end{aligned} \tag{11}$$

where $\hat{\pi}^j$ hypothesizes a distribution of A^j's actions. In addition to the methods mentioned above, other approaches, such as neural network-based methods like Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks, that map sequential data to probability distributions can also be implemented in $\mathcal{I}^i$.

After modeling the prediction of other agents' goals, we reverse the process to infer how they might predict our goals. In the LI-POMDP framework, the primary objective is to enable the legibility of the subject agent A^i's actions, making its goal g^i more discernible by others. To achieve this, A^i must be able to reason about how other agents perceive its goals below.

$$\mathcal{P}^i : \mathcal{O}_{t+1}^i \times \mathcal{A}_t^{-i} \longrightarrow \mathcal{B}_{t+1}^i \tag{12}$$

For example, I (A^i) am with my teammate (A^{-i}) in an intense Dota game. My possible goals g could be either to *attack* or to *retreat*. I would then take the actions $\mathcal{A}^i$, such as engaging in a fight, to achieve my goal, and I can observe my teammate's actions $\mathcal{A}^{-i}$ to predict whether he has understood my intention. If he moves forward to initiate the fight, I can infer that he has understood my offensive intention $\mathcal{B}^i$.

When there are more than two agents in the system, $\mathcal{B}^i$ is defined as a weighted sum in Eq. (13),

$$\hat{\mathbf{b}}_t^i = softmax(\sum_{j\in[1,N]}^{j\neq i} \omega_{i,j} \hat{\mathbf{b}}_t^{i|j}) \tag{13}$$

where $\hat{\mathbf{b}}^i \in \mathcal{B}^i$ is the belief of how other agents predict the goals of A^i, $\hat{\mathbf{b}}^{i|j}$ represents the probability distribution of A^j's prediction about A^i's goals, and $\omega_{i,j}$ is used to adjust the extent of legibility that A^i expresses to different agents, allowing A^i to prioritize the enhancement of the legibility towards those agents that are more beneficial to its collaborative efforts.

The implementation of $\mathcal{P}^i$ varies with the training paradigm. In *decentralized training*, subject agent A^i can utilize an estimator to infer $\hat{\mathbf{b}}^i$ based on the actions of other agents A^{-i}, assessing whether they have understood its intentions through the coordination in their actions. This process, known as *recursive reasoning* [43,44], can be exemplified as *I believe that you believe that I believe...* In a *centralized training* setting, $\mathcal{P}^i$ is more straightforward as A^i can directly communicate or query the other agents' understanding of its intentions (i.e., their estimation of its goals), allowing for more precise and computationally-friendly optimization of the legibility.

4.1.2. Making legible decisions

Due to the coupling of actions among agents, subject agent A^i must not only focus on accomplishing its sub-goal but also consider the intentions of other agents to achieve more effective coordination. The policy of A^i requires the concatenation input of the observation o_t^i, its goal $f_{\mathbf{1}}(g^i)$, and the estimation of the goals of other agents $\hat{\mathbf{b}}_t^{-i}$. Given the policy π^i, the subject agent A_i executes the action a_t^i. Upon the joint action $[a_t^i, a_t^{-i}]$, the environmental state is transited to the next state s_{t+1}, providing A^i with the new observation o_{t+1}^i and the raw reward r_{t+1}^i. At this point, given the observed actions of other agents a_t^{-i}, A^i employs the plan recognition $\mathcal{I}^i$ and recursive reasoning $\mathcal{P}^i$ to update $\hat{\mathbf{b}}_t^i$ and $\hat{\mathbf{b}}_t^{-i}$ (into $\hat{\mathbf{b}}_{t+1}^i$ and $\hat{\mathbf{b}}_{t+1}^{-i}$ respectively). Eventually, the raw reward r_{t+1}^i is transformed into an intrinsic reward $\tilde{r}_{t+1}^i$, incorporating the legibility critic, which is then used to update A^i's policy. The transformation of the reward function utilizing $\hat{\mathbf{b}}^{-i}$ is a key focus of this paper and will be elaborated in the following sections.

We utilize the *reward shaping* technique to enable the agent's behavior to be legible in solving LI-POMDP with the new objective function in Eq. (7). Specifically, we define the KL-divergence Gain (KLG), denoted by $\Delta D_{KL}(\mathcal{A}^i)$, as the difference of $D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i)$ before and after the action a_t^i is executed in the state s_t.

$$\Delta D_{KL}(a_t^i) = D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}_t^i) - D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}_{t+1}^i), t \geq 0 \tag{14}$$

In Eq. (14), $\Delta D_{KL}(a_t^i)$ quantifies the amount of the reduced uncertainty once the action a_t^i is executed. For instance, if the action a_t^i from A^i is very informative for A^{-i} to distinguish the plan, $\hat{\mathbf{b}}^i$ would converge to the subject agent's true goal $\mathbf{g}^i$ with the execution of a_t^i, and $D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i)$ would therefore reduce with it, and vice versa. Then, we add the legibility term $\Delta D_{KL}(a_t^i)$ to the original reward signal r_{t+1}^i with the parameter β to balance the scale.

$$\tilde{r}_{t+1}^i = r_{t+1}^i + \beta \Delta D_{KL}(a_t^i) \tag{15}$$

With the legibility incorporation, $\tilde{r}_{t+1}^i$ is used in the policy update of A^i through reinforcement learning. To summarize it, the legibility works in the following way: if a_t^i is helpful for $A^j \in A^{-i}$ to recognize the true goal of A^i, the KL-divergence between $\hat{\mathbf{b}}^i$ and $\mathbf{g}^i$ reduces, leading to a positive ΔD_{KL}, and eventually encouraging A^i to be more likely to choose a_t^i by increasing the reward.

In the end, for each agent, the problem can be simplified to an MDP and solved using single-agent reinforcement learning methods (with the environmental uncertainty already encapsulated in $(o^i \odot \mathbf{g}^i \odot \mathbf{b}^{-i})$). The policy update is performed using the tuple $[(o_t^i \odot \mathbf{g}^i \odot \mathbf{b}_t^{-i}), a_t^i, (o_{t+1}^i \odot \mathbf{g}^i \odot \mathbf{b}_{t+1}^{-i}), \tilde{r}_{t+1}^i]$. For instance, when using Q-Learning as the backbone algorithm, the policy is updated as follows:

$$Q(c_t^i, a_t^i) \leftarrow Q(c_t^i, a_t^i) + \alpha[\tilde{r}_{t+1}^i + \gamma \max_{a^{i\prime} \in \mathcal{A}^i} Q(c_{t+1}^i, a^{i\prime}) - Q(c_t^i, a_t^i)] \tag{16}$$

where α is the learning rate, $a^{i\prime}$ is a possible action of A^i, and c_t^i and c_{t+1}^i are the concatenations of $(o_t^i \odot \mathbf{g}^i \odot \mathbf{b}_t^{-i})$ and $(o_{t+1}^i \odot \mathbf{g}^i \odot \mathbf{b}_{t+1}^{-i})$ respectively.

4.2. The MAAL framework

In this section, we elaborate the MAAL framework in Fig. 2. It involves N agents, categorized as the subject agent A^i and other agents A^{-i}. Their shared objective, denoted as g^*, is divisible into the sub-goals: $g^* = g^i \cup g^{-i}$. A success is achieved when the subject agent A^i reaches $g^i \in \mathcal{G}^i$ and other agents A^{-i} attains $g^{-i} \in \mathcal{G}^{-i}$. The dashed lines colored by sky blue in Fig. 2 represent the operational framework from the perspective of subject agent A^i, equally applicable to other agents A^{-i}.

With the operational flow in Fig. 2, we elaborate the MAAL framework for N agents in Algorithm 1. We initialize the system at the start of each episode (Lines 2-4). Agents set their goals and reset their beliefs about other agents' goals, assuming a uniform distribution over each potential goal. At every time step t, A^i choose an action based on the policy $\pi^i(o_t^i \odot \mathbf{g}^i \odot \mathbf{b}_t^{-i})$ with exploration, such as epsilon-greedy [45] (Line 7, Gray block). Upon the influence of the joint actions $[a_t^0, \cdots, a_t^N]$, the environmental state is transited into the next state s_{t+1}. Meanwhile, the agents receive the rewards $[r_{t+1}^1, \ldots, r_{t+1}^N]$ and the observations $[o_{t+1}^1, \ldots, o_{t+1}^N]$ (Line 9, Orange dashed arrow). After all the other agents A^{-i} have executed their decisions, A^i observes their actions $\mathcal{A}_t^{-i}$ and conducts $\mathcal{I}^i$ and $\mathcal{P}^i$ to update $\hat{\mathbf{b}}^{-i}$ and $\hat{\mathbf{b}}^i$ (Line 11 and 12, Gold block). Subsequently, the agent A^i calculates ΔD_{KL} (Line 13, Pink block) and derives the intrinsic legibility reward $\tilde{r}_{t+1}^i$ (Line 14, Cyan block). Finally, a complete transition $[(o_t^i \odot \mathbf{g}^i \odot \mathbf{b}_t^{-i}), a_t^i, (o_{t+1}^i \odot \mathbf{g}^i \odot \mathbf{b}_{t+1}^{-i}), \tilde{r}_{t+1}^i]$ is derived for the immediate policy update (on-policy) or stored in the replay buffer (off-policy) in Line 15.

In summary, the MAAL framework facilitates mutual understanding and cooperation among multiple agents by enabling them to recognize and adapt to others' goals. Through a combination of plan recognition and belief estimation, agents can effectively collaborate towards shared objectives while enhancing the legibility of their actions so as to improve their performance.

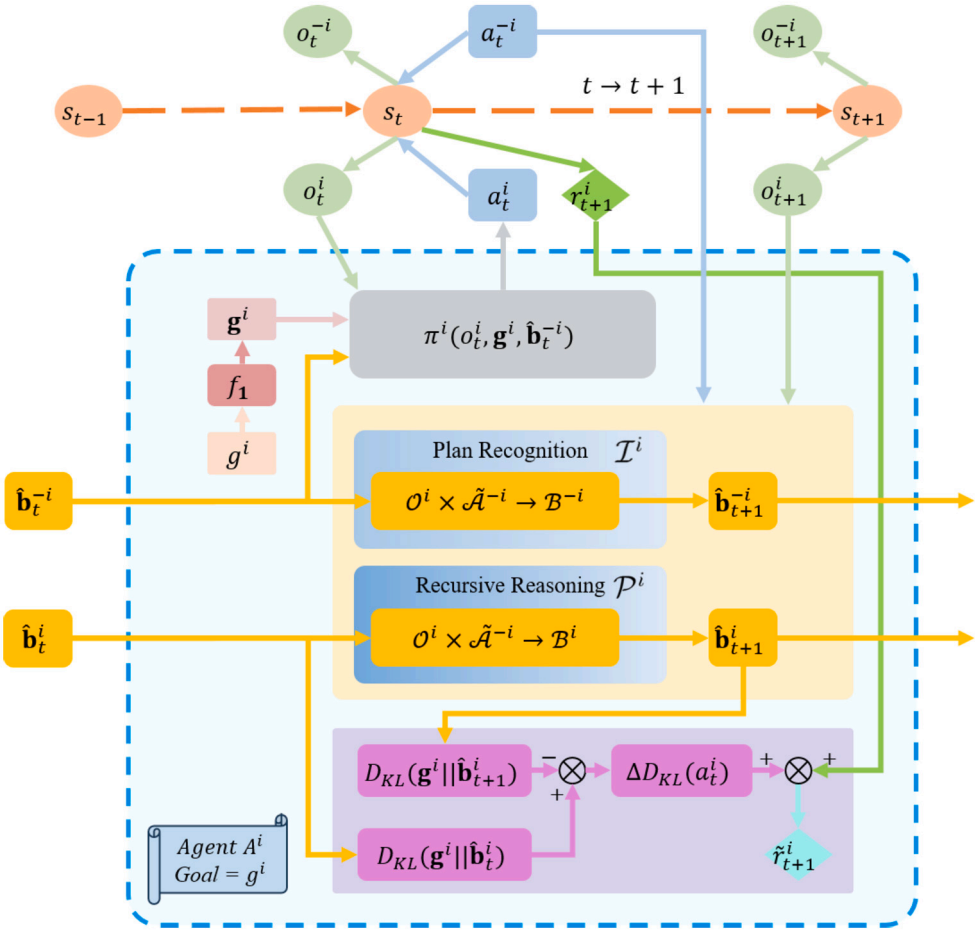


Fig. 2. The new framework of Multiagent Active Legibility (MAAL) is presented from the single-agent perspective (a subject agent A^i).

5. Completeness analysis

In this section, we first define the completeness in Definition 1, and analyze the MAAL reward shaping from the perspective of a single agent. Specifically, we examine if an agent can reach a sub-goal state under the original reward function, and if, using MAAL for the reward shaping, it satisfies Proposition 1, ensuring that the agent with enhanced legibility can also reach the target state. We first define the completeness of a policy π below.

Definition 1. Let s^0 represent the initial state, s^* the goal (absorbing) state, and $P_\pi^t(s^i, s^j)$ denote the probability of an agent reaching s^j after t-step transitions from s^i. Then a policy π is complete, if for every initial state s^0, there exists a finite time t such that $P_\pi^t(s^0, s^*) = 1$.

Proposition 1. *Let any agent A^i with a target sub-goal g^i be given. The necessary condition for applying the reward shaping in MAAL to agent A^i without compromising the completeness of the policy as defined in Definition 1 is: at any two time t_1 and t_2, when the agent A^i is in the same state s, the observing agent's estimation of agent A^i's goal must be consistent, i.e. $\hat{\mathbf{b}}_{t_1}^i = \hat{\mathbf{b}}_{t_2}^i$*

Proof. We refer to the convergence analysis of reward shaping [46], which reveals that one way an agent's policy is incompleteness when agents repeatedly visit non-goal states in pursuit of shaping rewards, thereby hindering the task completion. To further clarify it, we introduce a straightforward example to illustrate the potential issues arising from the improper reward shaping in multiagent systems. Fig. 3 illustrates the scenario where the reward shaping is used to improve the agent's exploration efficiency. Although the purpose of using the reward shaping in this context differs from this paper (where MAAL uses the reward shaping to directly alter the final policy), it is still highly valuable for analyzing the completeness.

Fig. 3(a) illustrates the MDP consisting of four states: $\{s^0, s^1, s^2$, and $s^*\}$, in which the s^0 is the initial state, and the agent's objective is to reach the target state s^*. The reward function for each transition is indicated above by the arrows. As shown in the figure, the agent receives the reward of +100 only upon making the final transition. Hence, the agent may require extensive

Algorithm 1: Multiagent Active Legibility.

Input: N agents: $\{A^1, ..., A^N\}$, N LI-POMDP: $\{\mathcal{M}^1, ..., \mathcal{M}^N\}$, Legibility weight: $\beta \in \mathbb{R}^+$, Maximum train episodes: $M \in \mathbb{N}^+$, Maximum episode steps: $L \in \mathbb{N}^+$
Result: N agent policies: $\pi^1, ..., \pi^N$

while $episodes \leq M$ **do**
  Reset the environment to s_0;
  Each agent is assigned with a goal: $g^i, i = 1, ..., N$;
  Initialize $\hat{\mathbf{b}}_0^{-i}, \hat{\mathbf{b}}_0^i, i = 1, ..., N$;
  while $t \leq L$ **do**
    for *Agent* $A^i \in \{A^1, ..., A^N\}$ **do**
      Choose the action via ε-greedy algorithm: $a_t^i \leftarrow \pi^i(o_t^i \odot \mathbf{g}^i \odot \hat{\mathbf{b}}_t^{-i})$;
    end
    Environment takes the joint action $[a_t^0, ...a_t^N]$, transit to the next state s_{t+1}, feedback the rewards $[r_{t+1}^1, ...r_{t+1}^N]$ and the observations $[o_{t+1}^1, ...o_{t+1}^N]$;
    for *Agent* $A^i \in \{A^1, ..., A^N\}$ **do**
      Update the overall belief: $\hat{\mathbf{b}}_t^{-i} \xrightarrow{o_{t+1}^i \times \mathcal{A}_t^{-i}} \hat{\mathbf{b}}_{t+1}^{-i}$ via Eq. (10)
      Update the belief: $\hat{\mathbf{b}}_t^i \xrightarrow{o_{t+1}^i \times \mathcal{A}_t^{-i}} \hat{\mathbf{b}}_{t+1}^i$ via Eq. (13)
      Calculate the KL-divergence Gain: $\Delta D_{KL}(a_t^i) \leftarrow D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_t^i) - D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t+1}^i)$
      Calculate the legibility reward: $\tilde{r}_{t+1}^i \leftarrow r_{t+1}^i + \beta \Delta D_{KL}(a_t^i)$
      Update the policy π^i or store the transition to buffer with tuple: $[(o_t^i \odot \mathbf{g}^i \odot \mathbf{b}_t^{-i}), a_t^i, (o_{t+1}^i \odot \mathbf{g}^i \odot \mathbf{b}_{t+1}^{-i}), \tilde{r}_{t+1}^i]$;
    end
    $t \leftarrow t + 1$;
  end
  $episodes \leftarrow episodes + 1$;
end

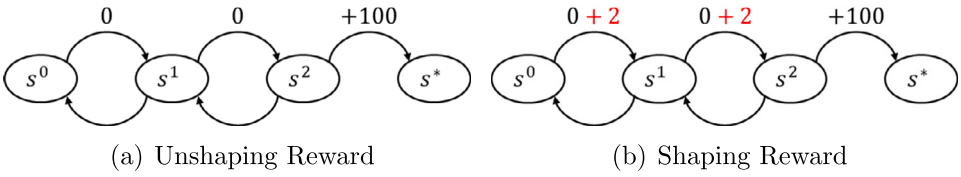


Fig. 3. An example of Reward Shaping in MDP.

exploration and trials to discover the state transition that completes the task. To enhance the agent's exploration efficiency, Fig. 3(b) introduces an additional reward of +2 (red numbers over the arrow) to the transitions $s^0 \to s^1$ and $s^1 \to s^2$ on top of the original reward function. This adjustment is intended to guide the agent towards the states s^1 and s^2, thereby improving its exploration efficiency. However, this reward function introduces a new problem: if the MDP has an infinite duration, the agent is to continuously cycle between s^0, s^1, and s^2 and accumulate the rewards that exceed the reward for reaching the target state s^* via $s^0 \to s^1 \to s^2 \to s^*$. In such a scenario, the agent accumulates significant rewards but fails to achieve the initial goal, thus losing the completeness of the algorithm.

Let's be back to the transition loop with the legibility reward shaping in MAAL and assume that agent A^i has the goal g^i. Starting from the state s^0 at time $t = 0$, the agent reaches the state s^n after n time steps and then returns to s^0 at time $t = n + 1$, forming a loop as: $s_{t=0}^0 \to ... \to s_{t=n}^n \to s_{t=n+1}^0$. Eventually the discounted return received by agent A^i is:

$$\begin{aligned} R = &\sum_{k=0}^{n+1} \gamma^k r_k^i + D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=0}^i) - D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=1}^i) + \gamma(D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=1}^i) - \\ &D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=2}^i)) + \gamma^2(D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=2}^i) - D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=3}^i)) + \cdots + \\ &\gamma^n(D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=n}^i) - D_{KL}(\mathbf{g}^i || \hat{\mathbf{b}}_{t=n+1}^i)) \end{aligned} \tag{17}$$

Compared to the original reward signal $\sum_{k=0}^{n+1} \gamma^k r_k^i$, MAAL provides an additional reward signal ΔR_{KLG} to agent A^i:

$$\Delta R_{KLG} = R - \sum_{k=0}^{n+1} \gamma^k r_k^i \tag{18}$$

To simplify the computation, we set γ to 1, reducing ΔR_{KLG} to a *telescoping series* as follows:

$$\begin{aligned}\Delta R_{KLG} = D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=0}) \underbrace{-D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=1}) + D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=1})}_{=0} \\ \underbrace{-D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=2}) + D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=2})}_{=0} \underbrace{-D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=3}) + D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=3})}_{=0} \\ \ldots \underbrace{-D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=n}) + D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=n})}_{=0} - D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=n+1})\end{aligned} \tag{19}$$

After eliminating the intermediate terms, ΔR_{KLG} can be expressed as the difference between the first and the last term:

$$\Delta R_{KLG} = \underbrace{D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=0}) - D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=n+1})}_{=0} \tag{20}$$

where $\hat{\mathbf{b}}^i_{t=0}$ is the prediction distribution of other agents for A^i at time $t = 0$ when it is in the state s^0. $\hat{\mathbf{b}}^i_{t=n+1}$ is the prediction distribution of other agents for A^i at time $t = n + 1$ when it is in state s^0 and has the trajectory: $s^0_{t=0} \rightarrow \ldots \rightarrow s^n_{t=n} \rightarrow s^0_{t=n+1}$.

In summary, to ensure that the agent A^i receives a non-positive reward ΔR_{KLG}, the observer's model must have the same goal estimation of A^i in one state: $D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=0}) = D_{KL}(\mathbf{g}^i||\hat{\mathbf{b}}^i_{t=n+1})$ □

6. Experimental results

We evaluate the MAAL performance on specific problem domains, where goal recognition is demanded among multiple agents, through comparative experiments and ablation study. In Section 6.1, we first introduce the Lead-Follow Maze (LFM), a discrete maze scenario with two agents and four exits. To diversify the experimental environments, we propose the *simple navigation* scenario in Section 6.2, which expands upon the simple spread task from the *particle* environment, incorporating more agents and landmarks. The simple navigation, with its continuous state-action space, allows for a more comprehensive evaluation of MAAL in complex settings, testing its ability to enhance the action's legibility and improve agents' decision making. In Section 6.3, we construct a scenario based on the StarCraft Multiagent Challenge: 1c1s_vs_1sc, where two Protoss units, namely *Stalker* and *Colossus*, must cooperate to defeat the Zerg unit Spine Crawler to win the game. Unlike the previous domains, we do not set explicit goals in this environment. Instead, we only define the capacity of the goal libraries and assign sub-goals randomly to evaluate the accuracy and speed of plan recognition among agents in a complex environment.

We not only conducted ablation experiments to compare the performance difference with/without the introduction of legibility, but also tested the impact of various legibility weights. Furthermore, we compared the proposed approach with several mainstream MARL algorithms. Each algorithm is run 5 times, with the mean represented by a curve and the variance depicted by a light shadow in the plot. The algorithms included in the experiments are as follows:

- Independent Q-Learning (IQL) [1] is an extension of the Q-Learning algorithm for multiagent settings. Multiple agents are trained independently, each with its own policy. The agents interact with the environment and learn to maximize their individual expected rewards.
- Value-Decomposition Networks (VDN) [2] assume that a global value can be represented as a sum of the individual values of each agent and aim to capture the interdependency between agents by decomposing the global value function.
- QMIX [4] employs deep neural networks to learn the value functions and the mixing network, allowing for more expressive representations and approximation of complex value functions compared to VDN.
- Multiagent Variational Exploration (MAVEN) [47] is an improved algorithm of the QMIX that overcomes the low exploration efficiency due to the monotonicity constraint.
- Probabilistic Recursive Reasoning (PR2) [10] is a probabilistic framework that takes into account opponents' potential reactions to their own latent behaviors and then attempts to find the best response to optimize its own decision-making.
- Mutil-Agent Deep Deterministic Policy Gradient (MADDPG) [5] is an extension of the Deep Deterministic Policy Gradient (DDPG), where each agent maintains its local actor network to make decisions, learns policy from a centralized critic network, and takes the joint action and observation of all the agents as the input.

6.1. Lead-Follow Maze

We design the scenario introduced in Section 1: a follow-the-leader task in a maze, referred to as the Lead-Follow Maze (LFM). As illustrated in Fig. 4, LFM is a discrete grid maze where the leader agent corresponds to the red humanoid in the example, and the follower agent corresponds to the green humanoid. Both agents can observe each other's actions and positions. The maze contains four circular exits, labeled A, B, C, and D, which correspond to the doors in the example. At the beginning of the game, the leader is assigned a target exit, which is unknown to the follower. The objective of the game is for the leader to guide the follower to the designated target exit. Hence, the follower must infer the true target exit by observing the leader's trajectory and reach the target

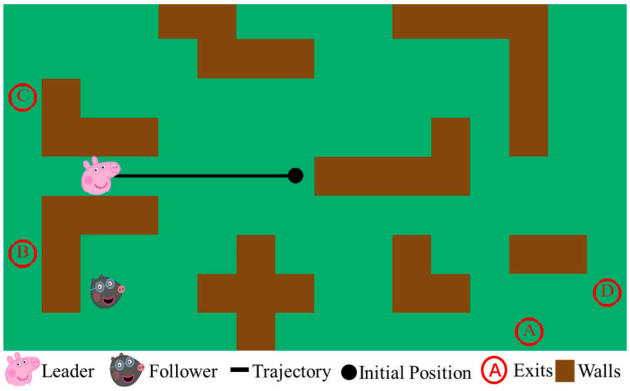


Fig. 4. The Lead-Follow Maze domain: a 10×16 grid maze with two agents and four exits.

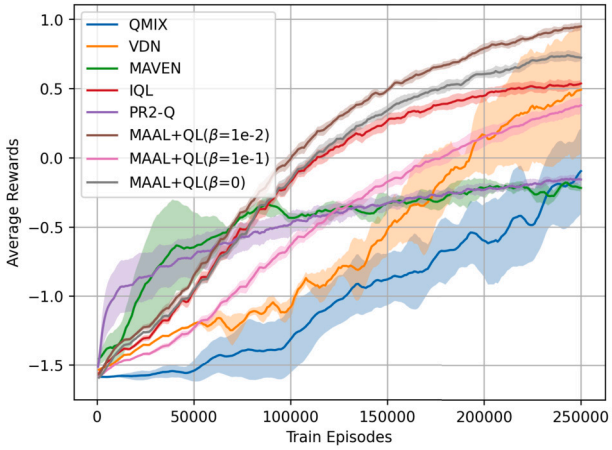


Fig. 5. Episode Reward in Lead-Follow Maze.

in time. Meanwhile, the leader can enhance the legibility of its actions to help the follower identify the target more quickly and accurately.

For example, the Leader is initialized at the black dot and moves left, crossing a corridor before descending to exit B. In this scenario, the Follower is unable to recognize the target before the Leader exits the corridor. Consequently, if the Follower follows the Leader step by step, it requires 11 steps to reach the target. However, if the Leader takes a legible action that allows the Follower to recognize the target earlier, the Follower only needs 5 steps to reach the exit, thus reducing the total motion cost and time consuming.

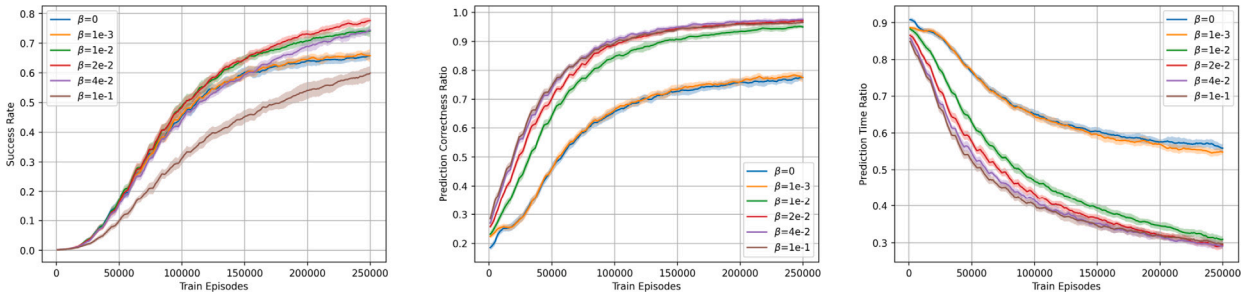
6.1.1. *Experimental settings*

Since both the state and action spaces are discrete, the leader and follower employ Q-Learning to learn their policies. In the LFM environment, the state space for both agents is defined as $\mathcal{S}$, which represents the coordinates of the leader and follower. For example, as shown in Fig. 4, $s = \{(2,4),(5,2)\}$, where $(2,4)$ indicates the coordinate of the leader, and $(5,2)$ represents the coordinate of the follower. For the Leader agent, the policy is fed with $[\mathcal{S} \odot g^*]$, where s is the state and g^* represents the target exit: $a^L \leftarrow argmax_{a' \in \mathcal{A}} Q^L([\mathcal{S} \odot g^*], a')$. As to the follower agent, its input is defined as $[\mathcal{S} \odot \hat{g}]$, where $\hat{g}$ is the follower's estimation of the true target, obtained from the Bayesian learning in Eq. (11). At the end of the episode, the follower utilizes the observation of leader's trajectory and the true target to update its plan recognition through parameter learning.

The action space for both agents is $\mathcal{A} = \{up, down, left, right, stay\}$, where each action moves one grid, and the state transition is deterministic. When the agents complete the task, they both receive a reward of $+1$. For each grid it moves, the agent receives the motion cost of -0.1.

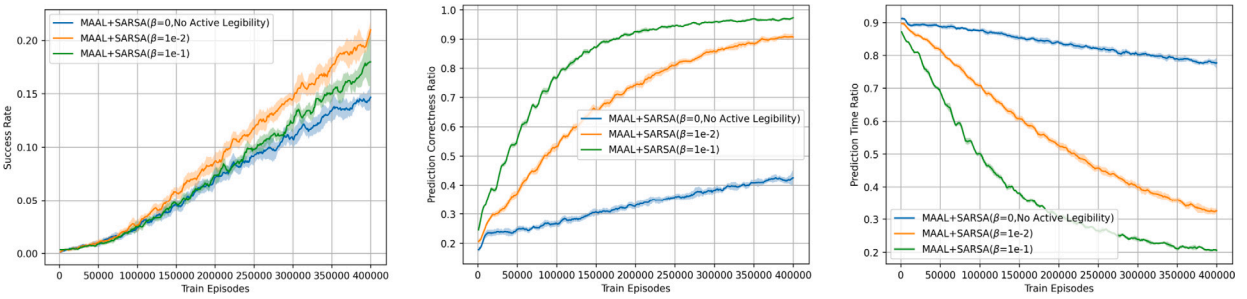
6.1.2. *Comparative experiments*

Fig. 5 presents the average rewards for the various algorithms, excluding shaped rewards. Interestingly, despite being one of the simplest algorithms, the IQL algorithm (red curve) performs remarkably well in this environment. Similarly, the VDN algorithm (orange curve) demonstrates good performance; however, it suffers from the high variance. Although QMIX (blue curve) is designed as an improved version of VDN, it performs poorly in this environment, significantly lagging behind VDN. This discrepancy may arise from the difficulty of training the non-linear combinations in QMIX, which do not adapt well to the environment. A similar pattern



(a) Success Rate (b) PCR (c) PTR

Fig. 6. Performance of the MAAL with Q-Learning and with different β values.



(a) Success Rate (b) PCR (c) PTR

Fig. 7. Performance of the MAAL with SARSA where the legibility applied.

is observed with the MAVEN algorithm (green curve). While it shows rapid improvement in the initial stage, its final performance only matches that of QMIX.

On the other hand, the MAAL+QL algorithm achieves the best performance in this environment. When the legibility weight β is set to 0.01, MAAL+QL (brown curve) outperforms all comparative algorithms. This outcome suggests that enhancing the legibility of an agent's policy improves the speed and accuracy of intention recognition between the agents, thereby boosting the collaborative performance. However, when β is increased to 0.1, the performance of MAAL+QL declines significantly. The higher reward shaping weight likely causes agents to overemphasize intention expression (i.e., improving legibility) at the expense of the task completion efficiency.

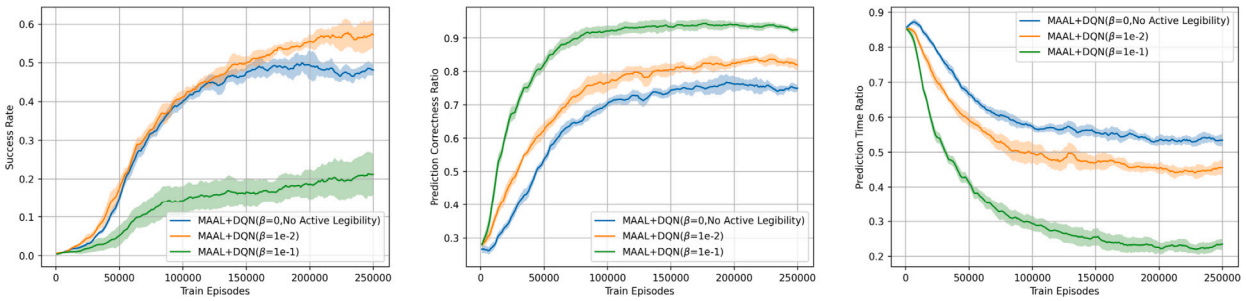
6.1.3. *Experiments on the legibility's impact*

We conducted comparative experiments with different legibility weights β under the same initial conditions. The experiments focused on the two metrics: Prediction Correctness Ratio (PCR) and Prediction Time Ratio (PTR). PCR refers to the accuracy of the follower's predictions of the leader' goals. For example, if within the training episodes 1000 to 2000, the follower correctly predicted the leader's goal in 415 out of 1000 episodes, the prediction correctness ratio at Episode 2000 is $415/1000 = 0.415$. PTR refers to the ratio between the number of steps required for an agent to correctly predict other agents' goals from the beginning of the episode to the length of the episode. A smaller PTR means that the agent can accurately predict other agents' goals earlier, suggesting higher legibility. For instance, if an observer correctly predicts the goal at Step 15 and maintains it to the end of the episode, with the total episode length being 50 steps, then the prediction time ratio for that episode is $15/50 = 0.3$.

We show the success rate in Fig. 6(a), PCR in Fig. 6(b) and PTR in Fig. 6(c) with different β values. It is noticed that the monotonic increase in PCR and decrease in PTR happen as β grows, which strongly indicates the improved legibility of the subject agent's behaviors. When the legibility is not applied, the follower's PCR is less than 70%, and requires nearly half of the journey before identifying the true goal of subject agent. Subsequently, as KLG exerts more influence on the reward signal, PCR climbs to nearly 100%, and the PTR is almost halved.

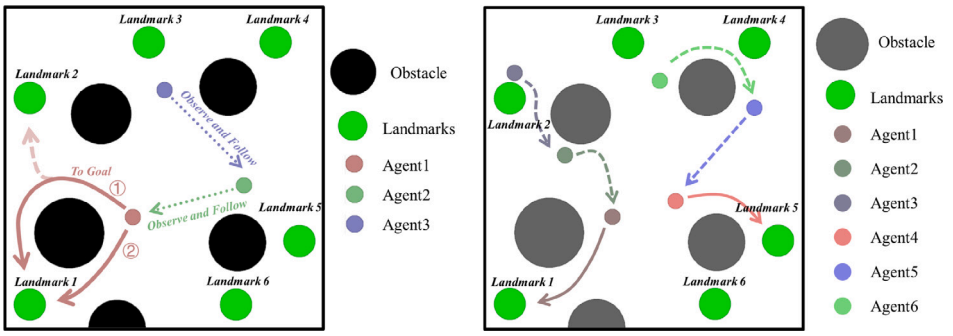
6.1.4. *MAAL beyond Q-learning*

As we mentioned before, MAAL stands upon the standard MDP and thus can be integrated into any single-agent reinforcement learning algorithm by solving MARL problems. We empirically study the legibility application in two different reinforcement learning algorithms. In this experiment, we incorporate MAAL with the State Action Reward State Action (SARSA) and Deep Q-Network (DQN) methods respectively, denoted as MAAL+SARSA and MAAL+DQN. To evaluate whether the legibility is helpful for multiagent learning, we set the parameter $\beta = 0$ in one of the experiments, i.e. removing the legibility weight in the reward function.



(a) Success Rate (b) PCR (c) PTR

Fig. 8. Performance of the MAAL with DQN where the legibility applied.



(a) 3 agents 1 goal (b) 6 agents 2 goals

Fig. 9. The simple navigation domain, where the agents navigate around the obstacles to the target landmarks. (*a*) illustrates a scenario with 3 agents and 1 goal, while (*b*) shows a more complex case with 6 agents and 2 goals.

Fig. 7(a) has shown, in MAAL+SARSA, the success rate with certain legibility (orange curve) reaches only 20%, but is still superior to those without legibility (blue curve) in 15%. Improving the legibility weight (green curve) causes a decrease in the success rate. The same phenomenon occurs to MAAL+DQN in Fig. 8(a). In MAAL+DQN, the usage of legibility raises the success rate by 10%. However, after increasing the legibility weight to 10^{-1}, the success rate of DQN sharply decreases, as DQN is more sensitive to rewards compared to SARSA and Q-Learning.

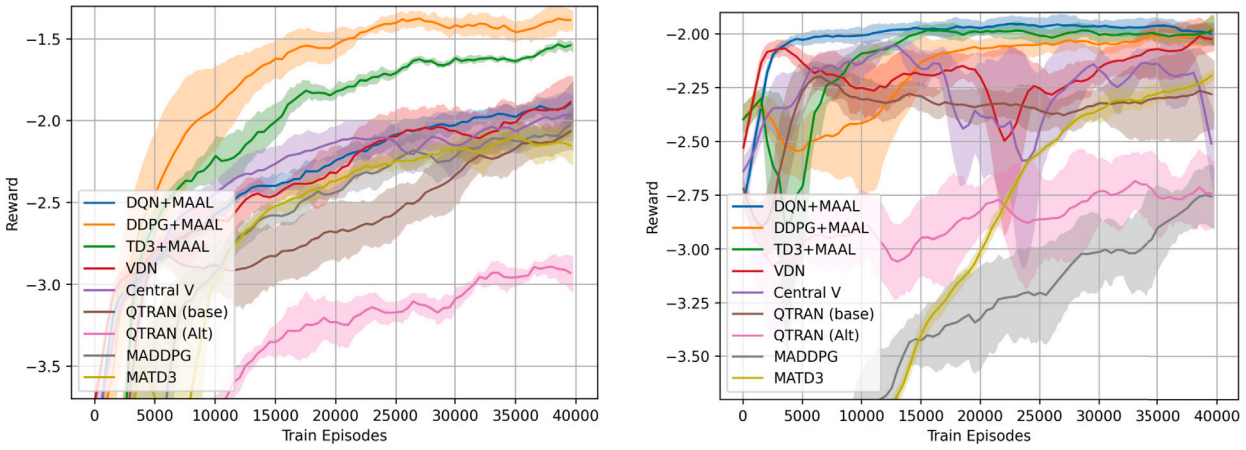
From Fig. 7(b) and Fig. 7(c), we can see that the legibility has a huge, positive impact on the goal identification in SARSA. Without MAAL, the Follower can only recognize the true target of the navigator in 40% episodes and obtain the correct results at almost the end of the episode. We also observe a similar pattern in MAAL+DQN from Fig. 8(b) and Fig. 8(c). We notice that the PTR has converged to approximately 25% in MAAL+QL, MAAL+SARSA, and MAAL+DQN, i.e., at least a quarter of the journey is necessary before recognizing the true goal.

In summary, the results from the three experiments demonstrate that, regardless of utilizing Q-Learning, SARSA, or DQN as the backbone algorithm, appropriately introducing legibility ($\beta > 0$) significantly improves the success rate compared to scenarios without legibility ($\beta = 0$). However, when the legibility weight is set too high ($\beta = 0.1$), the success rate declines across all the experiments. Excessive emphasis on the legibility leads the leader to prioritize predictable trajectories over optimal paths. Moreover, excessive legibility would reduce the weight of the original reward and incentivizes the agents to take paths that do not complete tasks but have $D_{KL} > 0$ to steal rewards. These factors jointly result in the decrease in overall task success rates, despite achieving high values in PCR and PTR. This phenomenon underscores the importance of finding a balanced trade-off between action legibility and task completion efficiency in multi-agent systems.

6.2. Particle simple navigation

Particle is a classic multiagent reinforcement learning environment proposed by OpenAI [5], where motion and collisions are simulated as real rigid-body collisions. Building upon Particle, we introduce a new scenario called *simple navigation*. Compared to LFM, the simple navigation scenario has a continuous state space with higher dimensions and more agents, making it much more complex than LFM.

As shown in Fig. 9, the scenario in Fig. 9(a) consists of three agents navigating around six landmarks and several immovable circular obstacles. Agent1 is assigned a target landmark unknown to agents2 and agent3. Agent2 can only observe the actions and trajectories of agent1, while agent3 can only observe those of agent2, forming an observation chain. Fig. 9(b) presents a more complex



(a) 3 agents 1 goal (b) 6 agents 2 goals

Fig. 10. Episode Reward in Simple Navigation.

setting with six agents and two distinct target landmarks, requiring coordination between agent1 and agents2 and 3, as well as agent4 leading agents5 and 6.

Similar to LFM, the legibility can significantly improve the agents' performance For instance, when agent1 aims to reach a target landmark, it faces two potential trajectories (represented by the solid red arrows in Fig. 9(a)). Although both the paths lead to the same destination, trajectory 1 creates ambiguity for the followers, as it could be perceived as leading toward an alternative landmark. In contrast, trajectory 2 eliminates such ambiguity, enabling the follower agents to predict the leader's target more accurately. In the specific configurations, such as when agent2 and agent3 are initially near the alternative landmark, trajectory 2 can reduce inefficient steps and improve the overall navigation efficiency.

6.2.1. *Experimental details*

We evaluate and compare the performance of MAAL with DQN, DDPG, and TD3 algorithms in the particle simple navigation. We begin by providing a detailed description of the experimental setup and parameters.

- **Input Regularization:** In this experiment, to deal with high-dimensional observation space, all the agents employ neural networks as policy, with vectorized $[\mathcal{O} \odot \mathcal{G}]$ as the network input, where $\mathcal{O}$ denotes the observation vector, and $\mathcal{G}$ represents the target landmark. For agent1 and agent3 (in the 6 agents, 2 goals scenario), their goals g^1 and g^3 are assigned at the beginning of the episode. And for other agents, such as agent2, $\hat{g}^2$ denotes the agent2's prediction of agent1's target and $\hat{g}^3$ represents the agent3's prediction of agent2's target (in one-hot encoded).
- **Plan Recognition:** We employ Long Short-Term Memory (LSTM) [48] to implement plan recognition. LSTM is a type of recurrent neural network (RNN) capable of learning long-term dependencies and sequences of data, making it well-suited for tasks involving time-series prediction and sequence classification. For example, agent2 uses its observations o^2 and the observed action of agent1 a^1 to predict the distribution of agent1's target: $LSTM(O^2_{t=0} \odot a^1_{t=0}, O^2_{t=1} \odot a^1_{t=1}, O^2_{t=2} \odot a^1_{t=2}, ...) \mapsto \hat{g}^2$. Similarly, agent3 uses the same kind of method to observe and predict the goal of agent2.
- **Parameters:** Based on the experience from the previous experiment, we set the legibility weight β for all three algorithms to 0.01 and the discount factor γ to 0.9.

6.2.2. *Comparative experiments*

The experimental results, illustrated in Fig. 10, demonstrate that integrating the MAAL reward shaping with various single-agent reinforcement learning algorithms significantly enhances their performance in the agents' navigation. Among all the methods, DDPG+MAAL (orange curves) achieves the highest rewards and the fastest converging speed in 3 agents domain, showing a steady and substantial improvement over the training episodes. The Twin Delayed Deep Deterministic Policy Gradient (TD3) [49] algorithm is an improved version of the DDPG algorithm, addressing some of its shortcomings by incorporating double Q-networks and delayed update mechanisms. Notably, the performance of TD3+MAAL (green curves) is not as well as DDPG+MAAL, likely due to the complexity introduced by TD3's double Q-networks and delayed policy updates. While these mechanisms reduce the overestimation bias and improve the stability in the single-agent scenarios, they might hinder the performance in multiagent settings where simpler and faster updates are crucial for effective coordination and interaction between agents. Although DQN+MAAL (blue curves) converges slower than TD3 and DDPG in the 3 agent domain, it shows a significant and substantial improvement in convergence speed and stability as the number of agents increases in the domain.

Regarding other traditional MARL algorithms that do not utilize the legibility and goal recognition, it can be observed that all VDN (red curve), Central V (purple curve), MADDPG (gray curve), and MATD3 (yellow curve) converge to around -2 at almost the same convergence rate. Although QTRAN-base (brown curve) also converges to a similar level, its convergence speed is significantly

(a) SMAC: 3s_vs_5z (b) SMAClite: 1c1s_vs_1sc

Fig. 11. The screenshots of SMAC scenarios. (*a*) 3 Stalker vs 5 Zealots in SMAC; and (*b*) Colossus and Stalker vs Spine Crawler in SMAClite.

slower than the others. Additionally, the alternative version of QTRAN (QTRAN-Alt, pink curves), which constructs transformed action-value functions in a different way, performs poorly, with results far below the baseline. In conclusion, these findings demonstrate the effectiveness of the MAAL framework in enhancing cooperation and goal recognition among the agents, leading to higher and more stable rewards in MARL.

6.3. StarCraft multiagent challenge

StarCraft Multiagent Challenge (SMAC) [50] is a multiagent reinforcement learning domain built upon the popular real-time strategy (RTS) game StarCraft II. In SMAC, agents learn to collaborate and select actions within a discrete action space, including moving up, moving down, moving left, moving right, attacking visible enemy 1, etc., based on field-of-view observations, to defeat enemy units. Due to the computational expense of the SMAC framework and its reliance on proprietary tools specific to StarCraft II for any meaningful modifications to the environment, we adopt the open-source, lightweight SMAClite framework [51] in our experiments. SMAClite allows us to easily design custom terrains and battle scenarios, facilitating a more thorough evaluation of the proposed MAAL framework in this study.

As illustrated in Fig. 11(b), we constructed the environment: 1c1s_vs_1sc, where three circles represent the units: the Protoss Colossus (green, labeled as “cls”), the Protoss Stalker (green, labeled as “stl”), and the Zerg enemy Spine Crawler (red, labeled as “CST”). The numbers below each circle indicate the unit’s health and shield values, while the numbers above represent the attack cooling time. In the map, the dark blue tiles denote the impassable obstacles.

6.3.1. Experimental settings

In this scenario, neither agent can defeat the enemy alone, making cooperation essential. For example, the Colossus leverages its high health and shield to attract enemy fire, creating opportunities for the Stalker to strike. Many existing studies solely focus on *micromanagement*, such as *focus fire*: coordinating agents to attack and eliminate enemy targets sequentially or *kiting*: arranging agents into formations based on armor types, luring enemy units into pursuit while maintaining a safe distance to avoid the damage.

As discussed earlier, this paper aims to improve the speed and accuracy of recognizing the intentions behind agents’ actions by the legibility. Hence, we define a goal library with a capacity of 3 for Colossus and Stalker: $|\mathcal{G}^C| = |\mathcal{G}^S| = 3$. However, unlike the previous experiments, we neither specify the exact meaning nor explicitly train the agents’ policies for each goal in $\mathcal{G}^C$ and $\mathcal{G}^S$. Instead, we adopt a random strategy, where the sub-goals are randomly sampled from the goal library and assigned to the agents at the beginning of each episode.

Since the domain features a continuous observation space and a discrete action space, we adopt DQN as the backbone policy. Each agent’s input is a vector $[\mathcal{O} \odot \mathcal{G} \odot \hat{\mathcal{G}}]$, where $\mathcal{O}$ represents the raw observation vector provided by the SMAClite environment, $\mathcal{G}$ denotes the agent’s sub-goal, and $\hat{\mathcal{G}}$ represents the agent’s estimation of its teammate’s sub-goal (both encoded as one-hot vectors). For the fairness, in the comparative experiments, the input vector is modified to $[\mathcal{O} \odot \mathcal{G} \odot \mathbf{0}]$ as well, where $\mathbf{0}$ is zero-padding of estimation for teammates.

6.3.2. Comparative results

The training performance of various algorithms on the 1c1s_vs_1sc is shown in Fig. 12. The proposed MAAL framework combined with DQN was evaluated under two settings: without legibility ($\beta = 0$, blue curve) and with legibility ($\beta = 0.05$, orange curve). These results were compared against other multiagent reinforcement learning algorithms, including VDN, QMIX, and COMA. The experimental results demonstrate that by episode 4000, all the VDN (green curve), QMIX (red curve), and COMA (purple curve) methods converge to an episode reward of 20 (battle win). During this process, both VDN and QMIX exhibit varying instability, while COMA exhibits the least variability. In the later stages of training, all algorithms except VDN maintain convergence. COMA demonstrates the highest stability, followed by QMIX, whereas VDN diverges and fails to consistently win games. This divergence is likely due to the overly simplistic value decomposition approach. In terms of the convergence speed, the three baseline algorithms exhibit similar rates. MAAL+DQN methods lag behind by approximately three times in the training episodes. This is attributed to the additional complexity faced by MAAL+DQN agents, as they must learn not only their own three behavioral patterns but also those

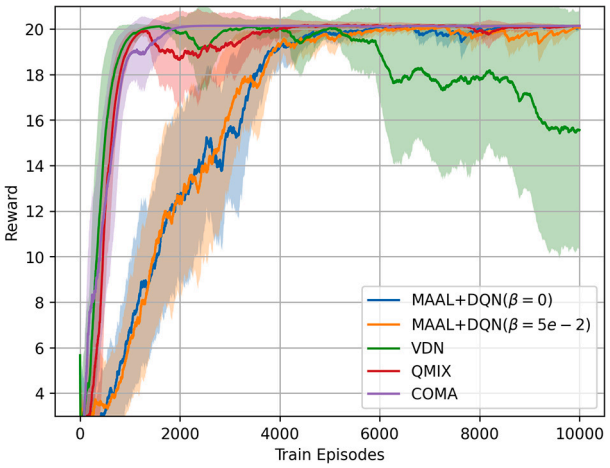
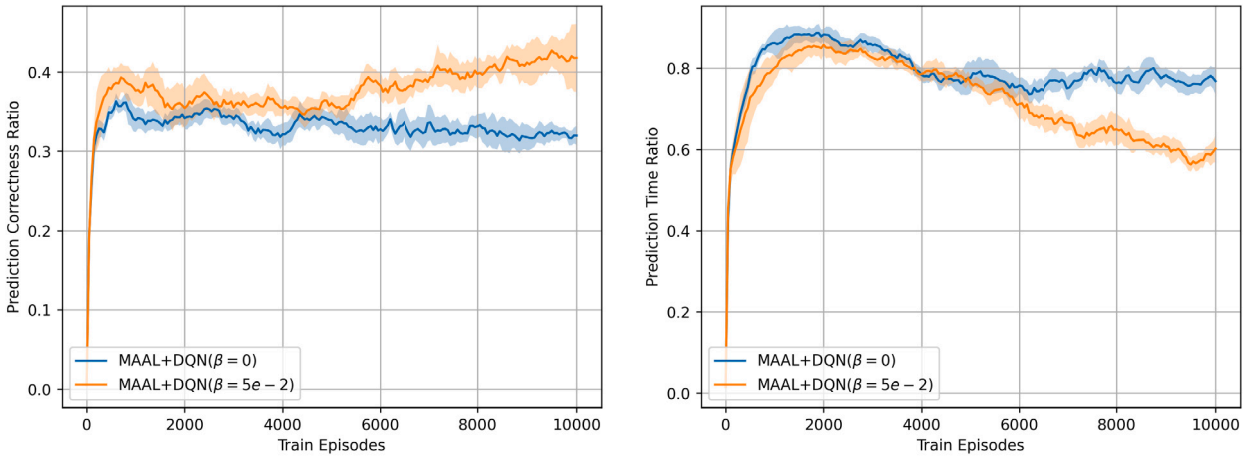


Fig. 12. Episode Reward in SMACLite: 1c1s_vs_1sc.



(a) PCR (b) PTR

Fig. 13. The peformance of PCR and PTR in 1c1s_vs_1sc.

of their teammates. This increases the number of behavior combinations by a factor of three compared to the baseline algorithms, resulting in a proportional increase in the number of training steps required.

6.3.3. The legibility's impact

Subsequently, we present the impact of legibility on PCR and PTR in Fig. 13. When using MAAL, the intention recognition accuracy reaches 41.7%, with a PTR of 0.61. In contrast, without legibility, the PCR is only 31.9%, while the PTR increases to 0.76. The results indicate that MAAL improves PCR by 10% and reduces the steps required for recognition by 15% in the SMAC domain. Although this improvement does not directly translate into better performance due to the random goal selection strategy, it can be hypothesized that integrating a more optimized goal-setting strategy would lead to significant performance gain.

6.4. Experimental summary

We summarize the experimental results and running times in Table 1. It shows that, by enhancing the legibility of agents' actions, we can effectively overcome the ambiguity and uncertainty in the plan recognition, which allows traditional single-agent RL algorithms to become more legible, and hence significantly reducing training cost. This is particularly evident in the lead-follow maze domain, where the policy is implemented using a Q-Table rather than a neural network, and plan recognition is achieved through Bayesian learning, resulting in lower computational complexity and faster convergence rates. As for the other problem domain, such as the simple navigator, MAAL does not bring significant advantages in the training time due to the high computational overhead of the plan recognition using LSTM. However, the MAAL algorithm still achieves the best results in the shortest time, demonstrating the applicability to the problems with high-dimensional continuous state and action spaces.

Table 1
The summary of experimental results includes the time and average rewards.

Domain	Methods	Time	Average Rewards
Lead-Follow Maze (250k episodes)	QMIX	10.3 h	-0.09 ± 0.3
	VDN	7.8 h	0.49 ± 0.46
	MAVEN	17.8 h	-0.22 ± 0.02
	IQL	0.2 h	0.54 ± 0.03
	PR2-Q	0.7 h	-0.14 ± 0.05
	QL+MAAL ($\beta = 0.01$)	1.3 h	$\mathbf{0.89 \pm 0.05}$
	QL+MAAL ($\beta = 0$)	1.3 h	0.72 ± 0.05
	SARSA+MAAL	0.8 h	-0.19 ± 0.07
	DQN+MAAL	12.8 h	-0.11 ± 0.13
Simple Navigator 3 agents 1 goals (40k episodes)	VDN	1.4 h	-1.88 ± 0.15
	Central V	2.4 h	-1.95 ± 0.12
	QTRAN (base)	3.6 h	-2.05 ± 0.13
	QTRAN (Alt)	3.7 h	-2.94 ± 0.10
	MADDPG	1.2 h	-2.01 ± 0.11
	MATD3	1.6 h	-2.14 ± 0.13
	DQN+MAAL	1.3 h	-1.88 ± 0.12
	DDPG+MAAL	0.9 h	$\mathbf{-1.39 \pm 0.06}$
	TD3+MAAL	1.1 h	-1.55 ± 0.03
Simple Navigator 6 agents 2 goals (40k episodes)	VDN	12.7 h	-2.02 ± 0.05
	Central V	6.0 h	-2.56 ± 0.19
	QTRAN (base)	11.6 h	-2.28 ± 0.16
	QTRAN (Alt)	15.4 h	-2.75 ± 0.15
	MADDPG	5.7 h	-2.77 ± 0.14
	MATD3	5.7 h	-2.19 ± 0.03
	DQN+MAAL	7.5 h	-2.00 ± 0.05
	DDPG+MAAL	6.4 h	$\mathbf{-1.97 \pm 0.06}$
	TD3+MAAL	6.4 h	-1.98 ± 0.07
SMAClite 1c1s_vs_1sc (10k episodes)	VDN	4.2 h	15.56 ± 5.21
	QMIX	2.4 h	20.00 ± 0.02
	COMA	1.9 h	20.00 ± 0.01
	DQN+MAAL ($\beta = 0$)	1.3 h	20.00 ± 0.1
	DQN+MAAL ($\beta = 0.05$)	1.3 h	20.00 ± 0.15

7. Conclusion and future work

In this paper, we propose a Multiagent Active Legibility (MAAL) framework to encourage agents to reveal their intentions as early as possible to achieve better collaboration with other agents. When combined with plan recognition, MAAL allows the agents to utilize a single-agent RL algorithm to achieve performance on par with, or even surpass, that of MARL methods in certain tasks. We employ the reward shaping technique in MAAL to make agent's actions more legible by reducing the ambiguity of its possible goals. Additionally, we designed two original experiments in discrete and continuous spaces: the Lead Follower Maze and Simple Navigation, both of which emphasize the speed and accuracy of the goal recognition between multiple agents. In these tests, we demonstrate the MAAL performance in those two scenarios compared to several MARL algorithms.

Although we have attained promising results in the environment we designed; however, the performance in other general multiagent learning environments remains unknown. In future work, we will investigate the integration of MAAL with task decomposition methods, which could generalize the application of the legibility-based opponent modeling multi-agent reinforcement learning paradigm to more complex tasks, thereby pushing the boundaries of decision-making algorithms in multi-agent systems. A set of benchmarks on testing the legibility's impact in MARL require more sophisticated design. It is clear that the legibility will contribute to explainable multiagent decision making from different perspectives, e.g. the agents by themselves or the observers outside the multiagent system. It would be very interesting to investigate their relations and possible unification.

CRediT authorship contribution statement

Yanyu Liu: Writing – original draft, Software, Methodology, Formal analysis, Data curation. **Yinghui Pan:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Yifeng Zeng:** Writing – review & editing, Supervision, Investigation, Conceptualization. **Biyang Ma:** Methodology, Conceptualization. **Prashant Doshi:** Writing – review & editing, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was also supported by the National Natural Science Foundation of China (Grant No. 62276168 and 62176225). Dr. Biyang Ma was supported by the Natural Science Foundation of Fujian Province, China (Grant No. 2022J05176). Dr. Yinghui Pan was supported by the Scientific Foundation for Youth Scholars of Shenzhen University, China (Grant No. 868-000001032177).

Data availability

Data will be made available on request.

References

[1] M. Tan, Multi-agent reinforcement learning: independent vs. cooperative agents, in: Proceedings of the International Conference on Machine Learning (ICML), 1993, pp. 330–337.
[2] P. Sunehag, G. Lever, A. Gruslys, W.M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J.Z. Leibo, K. Tuyls, et al., Value-decomposition networks for cooperative multi-agent learning based on team reward, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2018, pp. 2085–2087.
[3] K. Son, D. Kim, W.J. Kang, D.E. Hostallero, Y. Yi, Qtran: learning to factorize with transformation for cooperative multi-agent reinforcement learning, in: Proceedings of the International Conference on Machine Learning (ICML), in: PMLR, 2019, pp. 5887–5896.
[4] T. Rashid, G. Farquhar, B. Peng, S. Whiteson, Weighted qmix: expanding monotonic value function factorisation for deep multi-agent reinforcement learning, in: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 10199–10210.
[5] R. Lowe, Y.I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, in: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 6382–6639.
[6] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, S. Whiteson, Counterfactual multi-agent policy gradients, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 32, 2018, pp. 2974–2982.
[7] Y. Zeng, P. Doshi, Y. Pan, H. Mao, M. Chandrasekaran, J. Luo, Utilizing partial policies for identifying equivalence of behavioral models, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2011, pp. 1083–1088.
[8] T. Yang, J. Hao, Z. Meng, C. Zhang, Y. Zheng, Z. Zheng, Towards efficient detection and optimal response against sophisticated opponents, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2019, pp. 623–629.
[9] Z. Tian, Y. Wen, Z. Gong, F. Punakkath, S. Zou, J. Wang, A regularized opponent model with maximum entropy objective, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), vol. 28, 2019, pp. 602–608.
[10] Y. Wen, Y. Yang, R. Luo, J. Wang, W. Pan, Probabilistic recursive reasoning for multi-agent reinforcement learning, in: Proceedings of the International Conference on Learning Representations (ICLR), 2019.
[11] X. Yu, J. Jiang, Z. Lu, Opponent modeling based on subgoal inference, in: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), vol. 37, 2024, pp. 60531–60555.
[12] Z. Tian, R. Chen, X. Hu, L. Li, R. Zhang, F. Wu, S. Peng, J. Guo, Z. Du, Q. Guo, et al., Decompose a task into generalizable subtasks in multi-agent reinforcement learning, in: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), vol. 36, 2023, pp. 78514–78532.
[13] S. Miura, A.L. Cohen, S. Zilberstein, Maximizing legibility in stochastic environments, in: Proceedings of the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2021, pp. 1053–1059.
[14] Y. Liu, Y. Zeng, B. Ma, Y. Pan, H. Gao, X. Huang, Improvement and evaluation of the policy legibility in reinforcement learning, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2023, pp. 3044–3046.
[15] A. Dragan, S. Srinivasa, Generating legible motion, in: Proceedings of Robotics: Science and Systems, 2013.
[16] A.D. Dragan, K.C. Lee, S.S. Srinivasa, Legibility and predictability of robot motion, in: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2013, pp. 301–308.
[17] C. An, J. Ke, Prediction-based motion planning for uav interception, in: Proceedings of the IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), 2024, pp. 1273–1279.
[18] F. Shkurti, G. Dudek, Topologically distinct trajectory predictions for probabilistic pursuit, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 5653–5660.
[19] P.J. Gmytrasiewicz, P. Doshi, A framework for sequential planning in multi-agent settings, J. Artif. Intell. Res. 24 (2005) 49–79.
[20] P. Doshi, D. Perez, Generalized point based value iteration for interactive pomdps, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2008, pp. 63–68.
[21] P. Doshi, P.J. Gmytrasiewicz, Monte Carlo sampling methods for approximating interactive pomdps, J. Artif. Intell. Res. 34 (2009) 297–337.
[22] E. Sonu, P. Doshi, Scalable solutions of interactive pomdps using generalized and bounded policy iteration, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), vol. 29, Springer, 2015, pp. 455–494.
[23] Y. Zeng, P. Doshi, Q. Chen, Approximate solutions of interactive dynamic influence diagrams using model clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2007, pp. 782–787.
[24] B. Rosman, M. Hawasly, S. Ramamoorthy, Bayesian policy reuse, Mach. Learn. 104 (2016) 99–127.
[25] H. De Weerd, R. Verbrugge, B. Verheij, How much does it help to know what she knows you know? An agent-based simulation study, Artif. Intell. 199 (2013) 67–92.
[26] P. Doshi, X. Qu, A. Goodie, D. Young, Modeling recursive reasoning by humans using empirically informed interactive pomdps, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2010, pp. 1223–1230.
[27] F. Fotiadis, K.G. Vamvoudakis, Recursive reasoning with reduced complexity and intermittency for nonequilibrium learning in stochastic games, IEEE Trans. Neural Netw. Learn. Syst. 34 (11) (2022) 8467–8481.
[28] Y. Wen, Y. Yang, J. Wang, Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2021, pp. 414–421.
[29] S.V. Albrecht, P. Stone, Autonomous agents modelling other agents: a comprehensive survey and open problems, Artif. Intell. 258 (2018) 66–95.

[30] Z. Yi, nmode: neural memory ordinary differential equation, Artif. Intell. Rev. 56 (12) (2023) 14403–14438.
[31] R.M. Holladay, A.D. Dragan, S.S. Srinivasa, Legible robot pointing, in: Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, 2014, pp. 217–223.
[32] S. Nikolaidis, A. Dragan, S. Srinivasa, Based legibility optimization, in: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2016, pp. 271–278.
[33] M. Bied, M. Chetouani, Integrating an observer in interactive reinforcement learning to learn legible trajectories, in: Proceedings of the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2020, pp. 760–767.
[34] B. Busch, J. Grizou, M. Lopes, F. Stulp, Learning legible motion from human–robot interactions, Int. J. Soc. Robot. 9 (5) (2017) 765–779.
[35] M. Persiani, T. Hellström, Policy regularization for legible behavior, Neural Comput. Appl. 35 (23) (2023) 16781–16790.
[36] S. Bernardini, F. Fagnani, A. Neacsu, S. Franco, Optimizing pathfinding for goal legibility and recognition in cooperative partially observable environments, Artif. Intell. 333 (2024) 104148.
[37] X. Zhao, T. Fan, D. Wang, Z. Hu, T. Han, J. Pan, An actor-critic approach for legible robot motion planner, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 5949–5955.
[38] S. Miura, S. Zilberstein, Maximizing plan legibility in stochastic environments, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2020, pp. 1931–1933.
[39] S. Miura, S. Zilberstein, A unifying framework for observer-aware planning and its complexity, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), in: PMLR, 2021, pp. 610–620.
[40] S. Miura, S. Zilberstein, Observer-aware planning with implicit and explicit communication, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2024, pp. 1409–1417.
[41] M. Faria, F.S. Melo, A. Paiva, “Guess what I’m doing”: extending legibility to sequential decision tasks, Artif. Intell. 330 (2024) 104107.
[42] C.L. Baker, J.B. Tenenbaum, Modeling human plan recognition using Bayesian theory of mind, plan, activity, and intent recognition, Theory Pract. 7 (2014) 177–204.
[43] P.J. Gmytrasiewicz, E.H. Durfee, Rational coordination in multi-agent environments, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), vol. 3, Springer, 2000, pp. 319–350.
[44] H. De Weerd, R. Verbrugge, B. Verheij, Negotiating with other minds: the role of recursive theory of mind in negotiation with incomplete information, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), vol. 31, 2017, pp. 250–287.
[45] R. Sutton, A. Barto, Reinforcement learning: an introduction, IEEE Trans. Neural Netw. 9 (5) (1998) 1054.
[46] A.Y. Ng, D. Harada, S. Russell, Policy invariance under reward transformations: theory and application to reward shaping, in: Proceedings of the International Conference on Machine Learning (ICML), vol. 99, 1999, pp. 278–287.
[47] A. Mahajan, T. Rashid, M. Samvelyan, S. Whiteson, Maven: multi-agent variational exploration, in: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 7613–7624.
[48] H. Sak, A.W. Senior, F. Beaufays, et al., Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Proceedings of Interspeech, vol. 2014, 2014, pp. 338–342.
[49] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: Proceedings of the International Conference on Machine Learning (ICML), in: PMLR, 2018, pp. 1587–1596.
[50] M. Samvelyan, T. Rashid, C. Schroeder de Witt, G. Farquhar, N. Nardelli, T.G. Rudner, C.-M. Hung, P.H. Torr, J. Foerster, S. Whiteson, The starcraft multi-agent challenge, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2019, pp. 2186–2188.
[51] A. Michalski, F. Christianos, S.V. Albrecht, Smaclite: a lightweight environment for multi-agent reinforcement learning, arXiv preprint arXiv:2305.05566, 2023.