

Credit Assignment and Efficient Exploration based on Influence Scope in Multi-agent Reinforcement Learning

Shuai Han¹, Mehdi Dastani¹, Shihan Wang¹

¹Utrecht University

s.han@uu.nl, M.M.Dastani@uu.nl, s.wang2@uu.nl

Abstract

Training cooperative agents in sparse-reward scenarios poses significant challenges for multi-agent reinforcement learning (MARL). Without clear feedback on actions at each step in sparse-reward setting, previous methods struggle with precise credit assignment among agents and effective exploration. In this paper, we introduce a novel method to deal with both credit assignment and exploration problems in reward-sparse domains. Accordingly, we propose an algorithm that calculates the Influence Scope of Agents (ISA) on states by taking specific value of the dimensions/attributes of states that can be influenced by individual agents. The mutual dependence between agents' actions and state attributes are then used to calculate the credit assignment and to delimit the exploration space for each individual agent. We then evaluate ISA in a variety of sparse-reward multi-agent scenarios. The results show that our method significantly outperforms the state-of-art baselines¹.

1 Introduction

Multi-Agent Reinforcement Learning (MARL) has been widely applied in various fields in recent years, such as autonomous driving [Yeh and Soo, 2024], traffic signal control [Liu *et al.*, 2023b], and unmanned aerial vehicles [Kouzehgar *et al.*, 2023]. However, the success of MARL applications heavily relies on handcrafted reward functions to provide immediate feedback to agents. In sparse reward scenarios, MARL methods show low sample efficiency [Liu *et al.*, 2021a] or even fail to learn [Liu *et al.*, 2023a]. Inspired by solutions for single-agent sparse-reward domains [Andrychowicz *et al.*, 2017], goal-conditioned MARL has recently emerged as a promising approach by measuring individual goal achievement as intrinsic individual rewards for agents. With predefined subtasks [Iqbal *et al.*, 2022], sampled observations [Jeon *et al.*, 2022] or latent variables [Yang *et al.*, 2024] as goals, previous methods are able to learn competitive policies in sparse-reward MARL tasks.

However, there are still some open challenges for goal-conditioned MARL. Firstly, the multi-agent domain inherently amalgamates information from multiple agents [Lowe *et al.*, 2017; Samvelyan *et al.*, 2019], which brings challenges to automatically delimit dimensions/attributes from environmental observations or states for measuring individual goal achievement. Treating wrong information as an agent's goal can be harmful [Colas *et al.*, 2022]. For example, when training an agent to pick up an apple, it may not make sense to use the observed position of another agent as a goal. Secondly, the individual goal achievement of an agent may be affected by other agents, which means an agent may receive unstable or even wrong feedback because of the actions from other agents [Liu *et al.*, 2023a]. Thirdly, the state and joint action spaces of MARL increase exponentially with the number of agents [Yang *et al.*, 2024], which poses a challenge for exploring valuable states to identify the value of goals for specific tasks.

Aiming at solving the individual goal delimitation, credit assignment and exploration problems for goal-conditioned MARL mentioned above, we propose a novel method for sparse-reward MARL, which we call Influence Scope of Agents (ISA). ISA introduces the concept of influence scope for agents into multi-agent system, which can be efficiently and automatically calculated by measuring the mutual dependence between agents' discrete actions and state attributes/dimensions. This is done using the well-known information theoretic concept of mutual information between variables [Shannon and others, 1959]. Such influence scope of an agent delimits its individual goal space to provide succinct goal representation. Additionally, by identifying the joint influence scope of all agents, it can be automatically determined which segment of individual goal may be influenced by the team of agents. In credit assignment, the agent will not be rewarded from the segment if its current action cannot influence this segment. Moreover, the influence scope is also used to downscale the individual exploration space by excluding the dimensions/attributes that cannot be influenced by this agent, thereby improving the efficiency of exploration.

We verify the performance of our method on multiple tasks of challenging multi-agent sparse-reward environments. The results show that our method significantly outperforms the state-of-the-art methods in terms of both sample efficiency and final performance. Ablation experiments demonstrate the

¹The code is open-sourced at: <https://github.com/shan0126/ISA>

effectiveness of the proposed credit assignment and exploration methods based on influence scope. We also show the interpretability of ISA on the credit assignment among agents during training.

2 Related Work

Credit assignment in sparse-reward domains. In MARL, credit assignment is typically achieved by estimating the mixing value function [Rashid *et al.*, 2018; Son *et al.*, 2019] or learning a centralized critic [Wang *et al.*, 2021; Foerster *et al.*, 2018] to decompose the team reward to individual agents. In the sparse-reward environments where value functions and critics are difficult to learn, methods such as CM3 [Yang *et al.*, 2020] and MASER [Jeon *et al.*, 2022] introduce state or observation as individual goals to provide intrinsic reward to assist credit assignment. Besides, ALMA [Iqbal *et al.*, 2022] assigns credit by learning the assignment of goals. Building on the methods using individual goals to provide intrinsic rewards in sparse-reward domains [Jeon *et al.*, 2022; Colas *et al.*, 2022], this work further studies the representation of individual goals in MARL, proposes the individual goal representation based on influence scope of agent, and introduces a novel credit assignment method among agents based on the overlap of their influences.

Information-theoretic exploration or coordination. Information theoretic methods are often used to provide MARL with quantification [Wang *et al.*, 2020a; Li *et al.*, 2022]. In sparse-reward environments, these methods are particularly important. Empowerment-based methods [Salge *et al.*, 2014; Dai *et al.*, 2023] use mutual information to measure the controllable predictable consequences to construct the exploration bonus. With information entropy, CMAE [Liu *et al.*, 2021b] encourages agents to explore the states with less changes. EITI [Wang *et al.*, 2020b] quantify influence of one agent’s behavior on the reward of other agents via mutual information for better exploration. LAIES [Liu *et al.*, 2023a] tracks problem of lazy agent by investigating causality in MARL. FoX [Jo *et al.*, 2024] quantifies formations in MARL via mutual information to enhance exploration. HMASD [Yang *et al.*, 2024] coordinate the skills of different agents by maximizing the mutual information between state and skills. Distinguishing from these works, the information-theoretic machinery in our work is used to automatically determine the influence scope of agents on the state. And based on the influence scope, we enhance credit assignment and exploration efficiency. A more detailed description of related work can be found in Appendix A.

3 Preliminaries

Dec-POMDP. The fully cooperative MARL problem is described as a decentralized partially observable Markov decision process (Dec-POMDP) [Oliehoek and Amato, 2016], which is defined as a tuple $G = \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{O}, \Omega, R, \gamma \rangle$, where \mathcal{I} is the set of N agents, \mathcal{S} is the global state space of the environment, $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_N$ is the joint action space and \mathcal{A}_i is the action space of an individual agent i , $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability function, Ω is the observation space, $\mathcal{O} : \mathcal{S} \times \mathcal{I} \rightarrow \Omega$ is the

observation function, R is the shared reward function, and $\gamma \in [0, 1]$ is a discounted factor. When interacting with the environment, each agent i draws observation $o_i \in \mathcal{O}(s, i)$, where $s \in \mathcal{S}$ denotes the current global state. Then, each agent i samples its action $a_i \in \mathcal{A}_i$ with a stochastic policy $\pi_i : \mathcal{T}_i \times \mathcal{A}_i \rightarrow [0, 1]$ where $\mathcal{T}_i = (\Omega \times \mathcal{A}_i)^* \times \Omega$ represents the trajectory of agent i where $(\Omega \times \mathcal{A}_i)^*$ represents the Kleene closure on $\Omega \times \mathcal{A}_i$. After executing the joint action $\mathbf{a} = [a_1, \dots, a_N]$, the system transitions to a next state $s' \in \mathcal{S}$ and receives a shared reward r from R . The target of fully cooperative MARL is to learn the team policy to maximize the expected accumulated reward. In this work, we particularly consider the sparse-reward setting where the nonzero reward is not given to agents’ actions in every step but only when certain conditions are met [Yang *et al.*, 2024; Jo *et al.*, 2024]. Our methods follow the Centralized training & decentralized execution (CTDE) paradigm [Rashid *et al.*, 2018] of MARL where global information is available in training, but in execution, only local information is available.

Mutual information. Mutual information quantifies dependence between two variables, which is widely used in MARL to assist in policy learning. Given the probability distributions $p(x)$, $p(y)$ and the joint probability distributions $p(x, y)$ of two variables X and Y , the mutual information is:

$$I(X; Y) = \sum_{x, y} p(x, y) \log \frac{p(x)p(y)}{p(x, y)} = H(X) - H(X|Y) \quad (1)$$

where $H(\cdot)$ and $H(\cdot|\cdot)$ represent the entropy and conditional entropy respectively, and x and y are values from the range of variables X and Y , respectively. Equ. (1) can be read as how much knowing Y reduces uncertainty about X . The conditional mutual information is introduced when the information gain between two variables X and Y is conditioned upon a third variable Z , which can be calculated by:

$$I(X; Y|Z) = \mathbb{E}_z[I(X; Y|Z = z)] \quad (2)$$

where $I(X; Y|Z)$ is the mutual information of X and Y conditioned on Z and z represents a certain value of Z .

Additional notations. We now introduce some specific notations in this work. Given a Dec-POMDP, the state $s \in \mathcal{S}$ can be represented as a K -dimensional vector $s = [s^1, s^2, \dots, s^K]$, where $s^k \in \mathbb{R}$ denotes the value on k -th dimension of state s . Given a state vector s and an index set $D \subseteq \{1, 2, \dots, K\}$, we use $\text{Proj}_D(s) = (s^k)_{k \in D}$ to restrict the state vector to attributes indexed by D . For instance, for $D = \{3, 7\}$, $\text{Proj}_D(s) = (s^3, s^7)$ takes the attribute values of s at indexes $\{3, 7\}$.

4 Core Concepts

In this section, we introduce a novel concept called influence scope, which is grounded in an assumption that in a Dec-POMDP, an action always affect certain dimensions of the environmental state more significantly than other dimensions. We first leverage information theory to distinguish the dimensions of state that are significantly affected by actions, and then define the influence scope based on these dimensions. This assumption is aligned with the situations in many

practical applications. For instance, in autonomous driving where the gas pedal action of a vehicle affects the data from its own speed sensor, its speed will be included into the influence scope of this action. In contrast, in robotics tasks where a robot’s ‘walking’ action does not significantly affect the positions of other robots, these positions are excluded from this action’s influence scope.

We begin by describing how to distinguish which dimensions of states are affected by an action. Specifically, Conditioned on the actions of other agents \mathbf{a}_{-i} , we measure the mutual dependence between the (execution of) action a_i from agent i and the change on certain dimension k between the next state s' and the current state s ($\Delta s^k = s'^k - s^k$), which is denoted as $I(\Delta s^k; a_i | \mathbf{a}_{-i})$. This measures how much knowing the execution of a certain action a_i reduces uncertainty about the state change on k -th dimension given the action of other agents \mathbf{a}_{-i} . We use the value of this mutual information to quantify the influence of a certain action a_i on k -th dimension of state. If this influence exceeds a certain threshold δ , then we say the information on k -th dimension of state is influenced by this action a_i . In our method, $I(\Delta s^k; a_i | \mathbf{a}_{-i})$ is estimated from the expectation as follows.

$$I(\Delta s^k; a_i | \mathbf{a}_{-i}) = \mathbb{E}_{\tilde{\mathbf{a}}_{-i}} [I(\Delta s^k; a_i | \mathbf{a}_{-i} = \tilde{\mathbf{a}}_{-i})] \quad (3)$$

where $\tilde{\mathbf{a}}_{-i}$ represents specific values of variable \mathbf{a}_{-i} . When computing Equ. (3), we need to calculate the mutual information under different specific values $\tilde{\mathbf{a}}_{-i}$ of \mathbf{a}_{-i} and then take the expectation. If agents take random actions to collect interaction data, it will lead to an exponential variety of combinations for \mathbf{a}_{-i} and thus may cause the calculation in Equ. (3) to be intractable. In our practice, to calculate Equ. (3) for agent i , we randomly sample multiple combinations for \mathbf{a}_{-i} to calculate the average and to estimate the expected value. As the sample size grows large, this average provides an unbiased estimate of the expected value. Besides, when estimating the probability distributions of Δs^k and a_i , we first discretize the continuous variable Δs^k with equal width binning [Kraskov *et al.*, 2004] and convert a_i to a binary truth value where 1 represents the current action of agent i is a_i and 0 represents the current action of agent i is not a_i . Thus, our work only involves the estimation of mutual information for one-dimensional variables.

After measuring the influence of a action a_i of agent i on specific dimension of states by $I(\Delta s^k; a_i | \mathbf{a}_{-i})$, we can define the influence scope of a_i as follows.

Definition 1 (Influence scope of action). *Given a Dec-POMDP, the influence scope $D(a_i)$ for action a_i of agent i is an index set including all the dimension indices of the state that are influenced by this action, which is denoted by:*

$$D(a_i) = \{k \mid I(\Delta s^k; a_i | \mathbf{a}_{-i}) > \delta, s, s' \in \mathcal{S}\} \quad (4)$$

where $\delta \geq 0$ is a threshold.

Remark 1 (Fine-tunable influence scope of action): In our method, threshold δ is a hyperparameter to examine the level/degree of influences of an agent’s action on state dimensions. When $\delta = 0$, all dimensions of the state will be recognized as being affected by all actions. Conversely, when δ

is set to a very large value, the algorithm will perceive the influence scope of all actions as an empty set. Actually, which dimensions of the state are significantly affected by a specific action is an inherent property of the environment. In practice, we fine-tune δ to obtain suitable threshold across domains (More details are in Section 6).

Remark 2 (Credit assignment based on influence): The basic idea underlying Definition 1 is to use the influence scope to assign credits in multi-agent tasks. If a reward is caused by dimension k of state and $k \notin D(a_i)$, then a_i will not result in any reward. For example, by setting appropriate domain-dependent δ , agents can be aware that their ‘walking’ actions can significantly affect their location changes and their ‘pressing’ actions significantly affect whether a button is pressed. In this case, if the team of agents receive a reward because the button is pressed at some time step, then the agent performing ‘pressing’ will be assigned with this team reward and agents performing ‘walking’ will not be.

In MARL, the basic unit for receiving rewards and learning policies is the agent. Therefore, we define the influence scope of an agent based on the combined influence of all its actions.

Definition 2 (Influence scope of agent). *Given a Dec-POMDP and influence scope of actions $D(a_i)$ for all $a_i \in A_i$, the influence scope of agent i is an index set including all the dimension indices of the state that are influenced (for the given δ) by this agent, which is denoted by: $D_i = \cup_{a_i \in A_i} D(a_i)$.*

Definition 2 provides the influence scope of an agent through an index set, indicating which dimensions/attributes of the state are affected by the agent’s actions. This influence scope of agent in MARL is helpful for deriving the goal representation of individual agent. In single-agent tasks, desired states can simply be considered as the representation of goals because there is only one agent being controlled to influence the state [Sutton *et al.*, 2011]. However, a desired state in multi-agent tasks inherently amalgamates information influenced by multiple agents. We propose to distinguish this information based on agents’ influence scopes and use them as the representations of individual goals.

Definition 3 (Global goal and individual goal). *Given a Dec-POMDP and the influence scope D_i of agent i , a global goal $g = [s^1, \dots, s^K] \in \mathcal{S}$ is a vector from the K -dimensional state space. Given a global goal $g \in \mathcal{S}$, an individual goal g_i for an agent i is a projection $\text{Proj}_{D_i}(g) = (s^k)_{k \in D_i}$ that takes the value of the input vector g only at the indices given by D_i .*

The goal representation in reinforcement learning is closely tied to the intrinsic reward computation that measures the goal achievement [Colas *et al.*, 2022]. Following Definition 3, we propose to take the values of state attributes given by D_i as agent i ’s goal to provide (intrinsic) stimulus reward, which can be more efficient to stimulate agent i ’s behavior learning. This design is aligned with, for example, the concept of stimulus and reward in biology, where stimulus refers to environmental changes or signals that influence the actions of a living organism [Berridge and Robinson, 2003]. Note that D_i denotes the dimensions of the environment that have mutual dependence with (and influenced by) the actions of agent i .

Remark 3: (Non-conflicting individual goals): When the influence scopes of different agents overlap, their individual goals based on influence scope may in principle conflict in terms of the overlapping part. However, Definition 3 guarantees that agents always determine the value of their individual goal from the same global goal, ensuring that the overlapping parts of their individual goals are always consistent. For example, when the state value of a switch (on or off) is jointly influenced by two agents, their individual goals regarding this on/off value might differ. One agent might aim to have the switch off, while another aims to have it on. This leads to conflicting individual goals. However, since the on/off value from a given global goal is uniquely determined, the on/off value on different individual goals derived from this global goal is also uniquely determined. This design prevents conflicting goals.

Remark 4: (Trainable environments): Because the reward of a given Dec-POMDP depends on part or all the dimensions of state, there always exists an index set $D' \subseteq \{1, 2, \dots, K\}$ where the values of these state dimensions determine the reward of the Dec-POMDP. The environment is trainable by ISA only if all dimensions in D' can be influenced by the behavior of at least one agent, i.e., $D' \subseteq \cup_{i \in \mathcal{I}} D_i$. In fact, this condition can always be satisfied because, according to Remark 1, all dimensions will be considered to be within the agent’s influence scope when $\delta = 0$. In this case, our approach is consistent with the classic approach of treating the entire state as the goal for agents [Colas *et al.*, 2022]. And when the dimensions of state that determine the reward in the given Dec-POMDP are inherently influenced by the agents, the information gains underlying in Definition 1 \sim 3 allows ISA to find the inherent influence scope in the given environment and the corresponding individual goals that determine the reward of Dec-POMDP.

In MARL, the individual goal of an agent may not be influenced by its action alone but by the joint actions of team agents. Therefore, an agent needs to understand which segments of its individual goal are jointly influenced by team agents and which segments are not. To this end, we introduce the following definition.

Definition 4 (Common segment and special segment). *Given a Dec-POMDP, the influence scope of agent D_i for all $i \in \mathcal{I}$ and a global goal $g \in \mathcal{S}$, the common segment g_i^c of agent i is defined as the segment of its individual goal that affected by all agents, which is given by $g_i^c = \text{Proj}_{D^c}(g)$, where $D^c = \cap_{i \in \mathcal{I}} D_i$. The special segment $g_i^{(i-c)}$ for agent i is defined as the segment of its individual goal that excludes the common segment, which is given by $g_i^{(i-c)} = \text{Proj}_{D^{(i-c)}}(g)$, where $D^{(i-c)} = D_i \setminus D^c$.*

According to Definition 4, each agent has the same common segment on their individual goal, i.e., $g_i^c = g_j^c$ for all $i, j \in \mathcal{I}$. Besides, segments g_i^c and $g_i^{(i-c)}$ on individual goal g_i are given by index sets D^c and $D^{(i-c)}$, which can be understood as joint influence scope of all agents and influence scope for agent i without the joint influence. Definitions 1 \sim 4 integrate the concepts about influence scope into MARL, forming the foundation of methods in the rest of this paper.

5 Algorithm

Building on the concepts introduced above, a general process of the proposed ISA is as follows:

Step 1. Obtain influence scope. ISA first collects transitions by interacting with the environment and then computes the influence scopes $D(a_i)$ and D_i for each $i \in \mathcal{I}$ and each $a_i \in A_i$ according to Definitions 1 and 2 based on these collected transitions.

Step 2. Explore global goal. With the influence scopes of agents D_i and their actions $D(a_i)$, ISA trains the exploration policies $\{\pi_i^e\}_{i \in \mathcal{I}}$ where $\pi_i^e : \mathcal{T}_i \times \mathcal{A}_i \rightarrow [0, 1]$, to discover a set of success states as global goals. This step is necessary because in ISA agents do not have prior knowledge about the values of success states in the given Dec-POMDP.

Step 3. Train goal-conditioned policies. With at least one explored g , ISA trains policies conditioned on individual goals g_i decomposed from g based on Definition 3. Specifically, ISA trains goal-conditioned policies $\{\pi_i\}_{i \in \mathcal{I}}$ where $\pi_i : \mathcal{T}_i \times \mathcal{G}_i \times \mathcal{A}_i \rightarrow [0, 1]$ and \mathcal{G}_i represents the individual goal space. By uniformly sample a global goal g among the discovered set of global goals for a whole episode, g_i , decomposed from g , is used to be a part of input of π_i and to generate intrinsic rewards to train π_i . After repeated sampling of g and sufficient training, agents can be trained towards achieving multiple global goals discovered in Step 2, which are guaranteed by the multi-goal reinforcement learning paradigm [Schaul *et al.*, 2015; Colas *et al.*, 2022]. After centralized training, individual goals $\{g_i\}_{i \in \mathcal{I}}$, decomposed from a sampled global goal g , will be deployed locally to enable the decentralized execution of $\{\pi_i\}_{i \in \mathcal{I}}$.

Using the influence scopes and goals obtained through Definitions 1 \sim 4, the rest of this section will explain in detail how to train the goal-conditioned policies $\{\pi_i\}_{i \in \mathcal{I}}$ and the exploration policies $\{\pi_i^e\}_{i \in \mathcal{I}}$.

5.1 Goal-conditioned Credit Assignment

To train the goal-conditioned policies $\{\pi_i\}_{i \in \mathcal{I}}$, we first draw inspiration from the intrinsic reward function measuring the goal achievement in single-agent domain [Pignatelli *et al.*, 2023]. However, in multi-agent scenarios where the influence of agents on the environment overlaps, the goal achievement of an agent may also be influenced by other agents, which brings challenge to measure the contribution of individual agents to the achievement of their goals. In order to address this problem, we design a novel goal-conditioned reward functions for MARL. Specifically, we first divide each individual goal into common and special segments according to Definition 4. When evaluating the behavior of an agent i , we first measure the impact of this behavior on these two segments as rewards separately. The reward from common segment is calculated by:

$$R_i^c(s, s' | g_i^c) = d(s^c, g_i^c) - d(s'^c, g_i^c) \quad (5)$$

where $s^c = \text{Proj}_{D^c}(s)$ and $s'^c = \text{Proj}_{D^c}(s')$ are the restricted vectors of current and next states on dimensions given by D^c , g_i^c is the common segment of individual goal given by Definition 4 and d is the distance metric function between

two vectors. According to Equ. (5), this reward function produces a positive gain when the current state changes in a direction close to g_i^c . In this paper we use the combination of the Euclidean and Hamming distances: $d(v_1, v_2) = d_E(v_1, v_2) + \lambda d_H(v_1, v_2)$, where v_1, v_2 are the input vectors, d_E is the Euclidean distance, d_H is the Hamming distance, and λ is the hyper-parameter factor. Similarly, the reward from special segment is calculated by:

$$R_i^{(i-c)}(s, s' | g_i^{(i-c)}) = d(s^{(i-c)}, g_i^{(i-c)}) - d(s'^{(i-c)}, g_i^{(i-c)}) \quad (6)$$

where $s^{(i-c)} = \text{Proj}_{D^{(i-c)}}(s)$ and $s'^{(i-c)} = \text{Proj}_{D^{(i-c)}}(s')$ are the restricted vectors of current and next states on dimensions given by $D^{(i-c)}$, and $g_i^{(i-c)}$ is the special segment of individual goal given by Definition 4.

With these rewards from two segments, we can more precisely assign credits to each agent based on the action influence to different segments. Specifically, given a transition (s, a, s') and a global goal g sampled from discovered success states, each agent i 's goal-conditioned reward is:

$$R_i(s, a_i, s' | g_i) = \begin{cases} r_i^c + \alpha_1 r_i^{(i-c)} & \text{if } D(a_i) \cap D^c \neq \emptyset \\ \alpha_1 r_i^{(i-c)} & \text{Otherwise} \end{cases} \quad (7)$$

where g_i is the individual goal decomposed from the global goal g , $r_i^c = R_i^c(s, s' | g_i^c)$, $r_i^{(i-c)} = R_i^{(i-c)}(s, s' | g_i^{(i-c)})$, and α_1 is a factor to scale the reward from special segment. According to Equ. (7), within a , if the action a_i of agent i can affect the common segment of individual goal (i.e., $D(a_i) \cap D^c \neq \emptyset$), the goal-conditioned intrinsic individual reward of agent i will be computed from both the common segment and the special segment. Otherwise, its intrinsic individual reward will only be computed from the special segment. In this way, the credit can be assigned among agents. Finally, goal-conditioned policies $\{\pi_i\}_{i \in \mathcal{I}}$ are trained with the combinations of intrinsic and environmental rewards: $r_i + \alpha_2 r$, where $r_i = R_i(s, a_i, s' | g_i)$, r is the environmental reward and α_2 is a scaling factor.

5.2 Influence Scope Counting for Exploration

To train the exploration policies $\{\pi_i^e\}_{i \in \mathcal{I}}$ for discovering the success states as global goal more efficiently, we draw inspiration from the counting-based exploration [Strehl and Littman, 2008; Tang *et al.*, 2017]. Those methods employ bonus reward by counting states to motivate agent to explore new states. However, in multi-agent domain, counting-based methods can suffer from high-dimensional state space as the number of agents increases [Yang *et al.*, 2024], since most states will only occur once [Tang *et al.*, 2017]. To address this problem, we propose to use the influence scope of agents to downscale the segments of the state being counted. In our methods, the reward used for encouraging towards common segment jointly influenced by all agents is calculated by:

$$R_{i+}^c(s') = 1 / \sqrt{N(\varphi(s'^c))} \quad (8)$$

where $N(\cdot)$ returns the count of input vectors and φ is a hash function. Every time a specific $s'^c = \text{Proj}_{D^c}(s'^c)$ is encountered in the multi-agent system, $N(\varphi(s'^c))$ is increased by

Algorithm 1 Influence Scope of Agents (ISA) .

```

1: Initialize random exploration policies  $\{\pi_i^e\}_{i \in \mathcal{I}}$ 
2: Initialize random goal-conditioned policies  $\{\pi_i\}_{i \in \mathcal{I}}$ 
3: Initialize goal buffer  $\mathcal{B}$ 
4: Calculate  $D(a_i)$ ,  $D_i$ ,  $D^c$  and  $D^{(i-c)}$  in Definition 1, 2 and 4
5: for Episode 1 to  $M$  do
6:   Reset  $Env$ 
7:   if  $\text{len}(\mathcal{B}) < L$  then
8:     Collect a trajectory with  $\{\pi_i^e\}_{i \in \mathcal{I}}$ 
9:     for  $(s, a, s', r) \in \text{trajectory}$  do
10:      Count  $\varphi(s'^{(i-c)})$  for each  $i \in \mathcal{I}$  and  $\varphi(s'^c)$ 
11:      Obtain rewards  $r_{i+}$  for each  $i \in \mathcal{I}$  with Equ. (10)
12:      Update  $\pi_i^e, \forall i \in \mathcal{I}$  with IPPO loss and  $r_{i+} + \beta_2 r$ 
13:     end for
14:   else
15:     Sample a state as global goal  $g$  from  $\mathcal{B}$ 
16:     Decompose  $g$  into  $\{g_i\}_{i \in \mathcal{I}}$  based on Definition 3
17:     Collect a trajectory with  $\{\pi_i\}_{i \in \mathcal{I}}$  and  $\{g_i\}_{i \in \mathcal{I}}$ 
18:     for  $(s, a, s', r) \in \text{trajectory}$  do
19:      Calculate rewards  $r_i$  for each  $i \in \mathcal{I}$  with Equ. (7)
20:      Update  $\pi_i, \forall i \in \mathcal{I}$  with IPPO loss and  $r_i + \alpha_2 r$ 
21:     end for
22:   end if
23:   if trajectory is success then
24:     Store terminated state into  $\mathcal{B}$ 
25:   end if
26: end for

```

one. Similarly, the reward used for encouraging an individual agent towards the special segment solely influenced by itself is calculated by:

$$R_{i+}^{(i-c)}(s') = 1 / \sqrt{N(\varphi(s'^{(i-c)}))} \quad (9)$$

Equ. (8) and Equ. (9) measure the novelty of the state restricted by influence scopes. The more novel the projection, the greater the bonus. They can be used to motivate agents to explore new states within their influence. Similarly to Equ. (7), the exploration bonus to individual π_i^e is defined as:

$$R_{i+}(s, a_i, s') = \begin{cases} r_{i+}^c + \beta_1 r_{i+}^{(i-c)} & \text{if } D(a_i) \cap D^c \neq \emptyset \\ \beta_1 r_{i+}^{(i-c)} & \text{Otherwise} \end{cases} \quad (10)$$

where $r_{i+}^c = R_{i+}^c(s')$, $r_{i+}^{(i-c)} = R_{i+}^{(i-c)}(s')$ and β_1 is the scaling factor. Finally, exploration policies $\{\pi_i^e\}_{i \in \mathcal{I}}$ are trained with the combinations of exploration and environmental rewards: $r_i + \beta_2 r$, where $r_{i+}^c = R_{i+}(s, a_i, s')$, r is the environmental reward and β_2 is a scaling factor.

5.3 Algorithm

We organize the pseudo-code of ISA in Algorithm 1. After the initialization in Lines 1~3, the influence scope will be computed in Line 4 before training. During training, $\{\pi_i^e\}_{i \in \mathcal{I}}$ is first trained to discover successful states in Lines 8~13. When enough goals are collected (i.e., $\text{len}(\mathcal{B}) \geq L$ where $\text{len}(\mathcal{B})$ represents the length of buffer \mathcal{B}), goal-conditioned policies $\{\pi_i\}_{i \in \mathcal{I}}$ will learn to solve the task in the given Dec-POMDP in Lines 15~21. During interaction, success states found by $\{\pi_i^e\}_{i \in \mathcal{I}}$ are stored in goal buffer \mathcal{B} in Lines 23~25. We use the IPPO loss [De Witt *et al.*, 2020] to train $\{\pi_i^e\}_{i \in \mathcal{I}}$

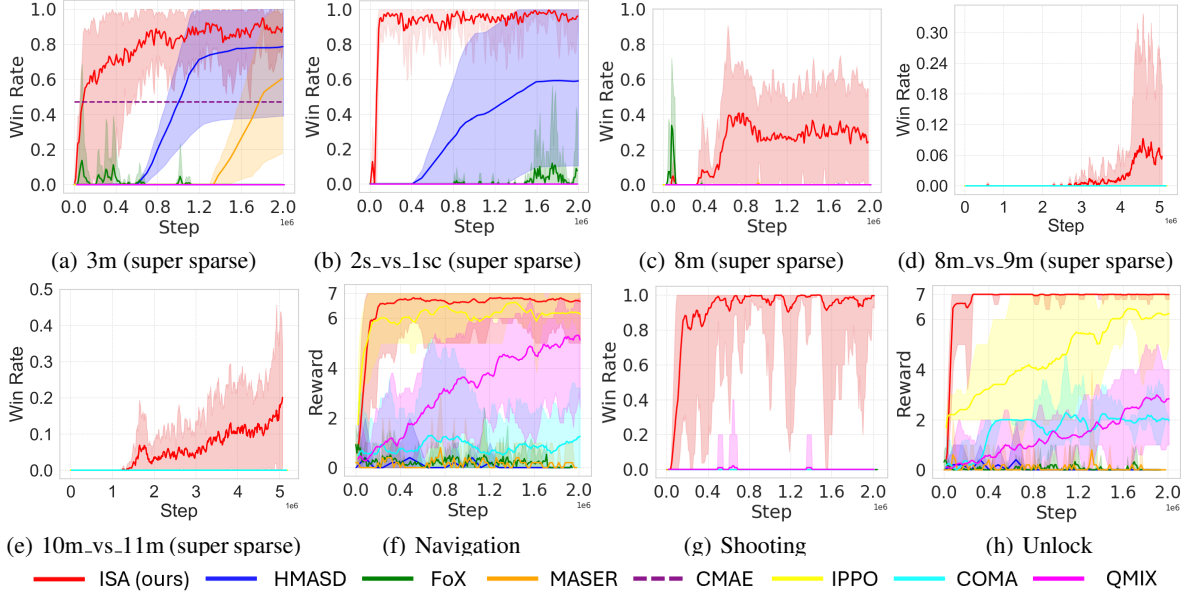


Figure 1: Learning curves on SMAC (with only +1/-1 reward) and MPE.

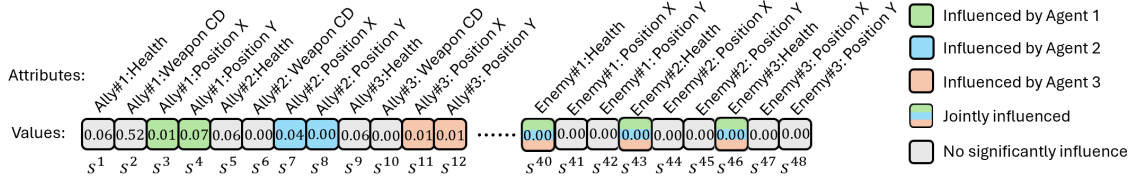


Figure 2: An illustration on decomposing individual goals from a global goal.

and $\{\pi_i\}_{i \in \mathcal{I}}$ separately with their corresponding rewards. We include a detailed version of pseudo-code and the description for the loss and network structure in Appendix B.

6 Experiments

Environment. We evaluate our method in two challenging benchmark domains with sparse-reward settings: (1) the starcraft multi-agent challenge (SMAC) [Samvelyan *et al.*, 2019]; and (2) the multiple-particle environment (MPE) [Lowe *et al.*, 2017]. SMAC is a real-time strategy game in which agents need to learn cooperative policies to eliminate the enemies. We focus on the super sparse setting of SMAC where a ‘+1’ reward will be given only when all enemies are eliminated and a ‘-1’ reward will be given only when all controlled agents die. This setting is very challenging because there are no immediate rewards in the environment.

In addition to SMAC, we also consider 3 tasks in MPE, i.e., Navigation, Shooting and Unlock. In the Navigation task, agents need to learn to occupy different landmarks where a ‘+1’ reward is given to the team when a landmark is occupied. In the Shooting task, agents need to shoot a target enough times and move towards specific positions where a ‘+1’ reward is given only when these subtasks finished. In the Unlock task, agents holds different keys to unlock the corresponding (multiple) locks where a ‘+1’ reward is given to the team when a lock is unlocked.

Baselines. Our baselines cover classical MARL methods (i.e., IPPO [De Witt *et al.*, 2020], QMIX [Rashid *et al.*, 2018], COMA [Foerster *et al.*, 2018]), methods with similar distance-based intrinsic rewards to ISA (i.e., MASER [Jeon *et al.*, 2022]), and state-of-the-art methods in sparse-reward MARL domains (i.e., CMAE [Liu *et al.*, 2021b], FoX [Jo *et al.*, 2024], HMASD [Yang *et al.*, 2024]). To the best of our knowledge, HMASD is the method with the best performance in the super-sparse SMAC domain.

Hyperparameters. We run ISA on 2.60 GHz AMD Rome 7H12 CPU. The hyperparameter settings for the learning part of ISA make reference to IPPO [De Witt *et al.*, 2020]. For the introduced hyperparameter δ , we fine-tune its value to find the workable range. We observed when $\delta \in [0.15, 0.45]$, the identification is correct for almost all actions across domains. We set $\delta = 0.3$ in all tasks and this works well. The scale factor λ for Hamming distance is 10 in 8m, 0 in Navigation and Unlock, and 50 in other environments. The scaling factors in Equ. (7) and (10) are $\alpha_1 = \beta_1 = 0$ for SMAC domain and $\alpha_1 = \beta_1 = 0.2$ for MPE domain. The scaling factors for environmental reward is $\alpha_2 = \beta_2 = 10$ in 8m, $\alpha_2 = \beta_2 = 0$ in 3m and 2s_vs_1sc, and $\alpha_2 = \beta_2 = 1$ in MPE domain. The number of transitions N to calculate influence scopes in Algorithm 1 is 10,000 in 8m and 2,000 in other environments. The length L in Algorithm 1 is 1 for all environments. More detailed descriptions about environments, baselines and hy-

hyperparameters are included in Appendix C.1~C.3.

Results. We compare our method with baselines on 6 environments from SMAC and MPE domains to validate the superiority of ISA. The results are shown in Fig. 1. The error bounds (i.e., shadow shapes) indicate the upper and lower bounds of the performance with 5 runs using different random seeds. Due to the delayed effect of actions in 2s_vs_1sc, we take into account the caused state changes in the next 2 steps of the current action when calculating the influence scope of actions. The results show that ISA significantly outperform the other methods in terms of both sample efficiency and learning performance.

Goal decomposition. We also perform an experiment to show the goal decomposition results of ISA. The result is shown in Fig. 2 where the squares represent a 48-dimensional success state $s = [s^1, s^2, \dots, s^{48}]$ explored from the 3m task. Taking this state as a global goal, individual goals of agents are decomposed based on the dimensions/attributes that they can influence. For instance, the individual goal of agent 1 is constituted by the green (special segment) and colorful (common segment) squares, which includes the information of its own position and the health of enemies.

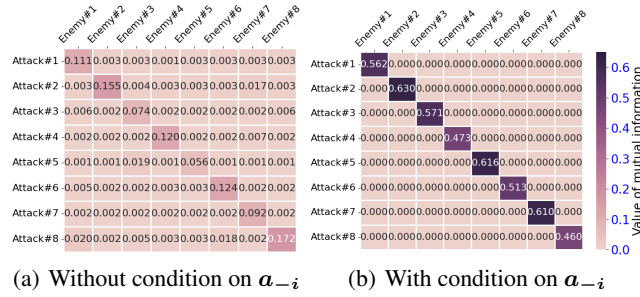


Figure 3: Heat maps of mutual information values.

Ablations. We perform ablations on the mutual information calculation conditioned on a_{-i} and not conditioned on a_{-i} in Equ. (3). The results are shown in Fig. 3. The values in Fig. 3 represent the values of mutual information. For example, the value in the third row and third column in Fig. 3(a) indicates mutual information between agent 1 performing action ‘attack enemy 3’ and the state change on the dimension that indicating ‘enemy 3’s health’ when not conditioned on a_{-i} . Correspondingly, the value in the third row and third column in Fig 3(b) represents the same mutual information but conditioned on a_{-i} . According to the results of Fig. 3, the influence on state of the agent’s actions can be distinguished more significantly when conditioned on a_{-i} .

Besides, we perform ablations to verify the contribution of proposed influence scope and the effectiveness of credit assignment in both Equ. (7) and Equ. (10). We compare the ISA with its ablative variants on 2s_vs_1sc in Fig. 4(a). ‘ISA w/o influence scope’ ablates the influence scope in ISA (preserving the count-based exploration over all state dimensions), which fails to learn due to the large exploration space. ‘ISA individual goal’ ablates the segmentation on individual goals (constructing the reward based on the distance between

current state and individual goals), which shows lower sample efficiency and instability because of the wrong credit assignment during both exploration and learning. ‘ISA w/o Equ. (7)’ ablates the judging process for credit assignment in Equation (7) based on the influence of action, which shows instability of policy learning due to wrong reward in training. ‘ISA w/o Equ. (10)’ ablates the judging process for credit assignment in Equation (10) in exploration, which shows lower sample efficiency to find the success states as global goals to start goal-conditioned policy learning.

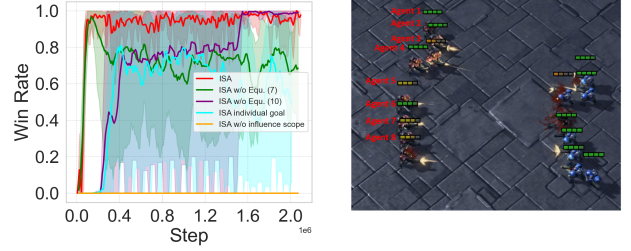


Figure 4: Ablations and interpretability for ISA

Interpretability. Our credit assignment based on the influence scope offers good interpretability. Based on the if-else rule in Equ. (7), we can interpret whether a specific action a_i of agent i has influence on the common segment D_c . For instance, when $D(a_i) \cap D^c = \emptyset$, the current action a_i has no influence on D_c , and as a result, no reward from common segment shall be assigned to agent i . Fig. 3(b) illustrates this scenario. At the time step of this screenshot, all agents are shooting to enemies except agent 7. This indicates that the current action of agent 7 does not contribute to the state changes delimited by D_c (the health of all enemy). Consequently, ISA ensures a fair credit assignment by awarding agent 7 less rewards than the other agents.

Besides the above experiments, we also verified the performance stability of ISA under different hyperparameters, the statistical significance of performance improvements, and the limited amount of the introduced time consumption. The results are shown in Appendix C.3, C.4, and C.5 respectively.

7 Conclusions and future work

In this paper, we propose ISA, an algorithm that improves both credit assignment and exploration in MARL. ISA measures the mutual information between agents’ actions and the state attributes/dimensions to identify the influence scope of agents. ISA use the influence scope to provide a precise and succinct representation for individual goals. Then, the credit assignment for the individual agent is determined based on the influence of its current action on its individual goal. Besides, a novel exploration method is proposed by restricting the state to be explored by agents to the attributes/dimensions of what they can influence, which improves the exploration efficiency. We show that in a variety of sparse-reward MARL environments, ISA significantly outperforms the state-of-the-art methods. In this work, goals are without hierarchies. In

future work, we are going to study the hierarchical goals based on influence scope and combine it with hierarchical MARL to further improve the efficiency of the algorithm.

References

- [Andrychowicz *et al.*, 2017] Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5048–5058, 2017.
- [Berridge and Robinson, 2003] Kent C Berridge and Terry E Robinson. Parsing reward. *Trends in neurosciences*, 26(9):507–513, 2003.
- [Colas *et al.*, 2022] Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: A short survey. *J. Artif. Intell. Res.*, 74:1159–1199, 2022.
- [Dai *et al.*, 2023] Siyu Dai, Wei Xu, Andreas Hofmann, and Brian Williams. An empowerment-based solution to robotic manipulation tasks with sparse rewards. *Autonomous Robots*, 47(5):617–633, 2023.
- [De Witt *et al.*, 2020] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.
- [Foerster *et al.*, 2018] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2974–2982, 2018.
- [Iqbal *et al.*, 2022] Shariq Iqbal, Robby Costales, and Fei Sha. ALMA: hierarchical learning for composite multi-agent tasks. In *Advances in Neural Information Processing Systems*, 2022.
- [Jeon *et al.*, 2022] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. MASER: multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International Conference on Machine Learning*, volume 162, pages 10041–10052, 2022.
- [Jo *et al.*, 2024] Yonghyeon Jo, Sunwoo Lee, Junghyuk Yeom, and Seungyul Han. Fox: Formation-aware exploration in multi-agent reinforcement learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 12985–12994. AAAI Press, 2024.
- [Kouzehgar *et al.*, 2023] Maryam Kouzehgar, Youngbin Song, Malika Meghjani, and Roland Bouffanais. Multi-target pursuit by a decentralized heterogeneous UAV swarm using deep multi-agent reinforcement learning. In *International Conference on Robotics and Automation*, pages 3289–3295, 2023.
- [Kraskov *et al.*, 2004] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.
- [Li *et al.*, 2022] Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E. Taylor, Wenyuan Tao, and Zhen Wang. PMIC: improving multi-agent reinforcement learning with progressive mutual information collaboration. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12979–12997, 2022.
- [Liu *et al.*, 2021a] Iou-Jen Liu, Unnat Jain, Raymond A. Yeh, and Alexander G. Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 6826–6836, 2021.
- [Liu *et al.*, 2021b] Iou-Jen Liu, Unnat Jain, Raymond A. Yeh, and Alexander G. Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 6826–6836, 2021.
- [Liu *et al.*, 2023a] Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and D. Zhang. Lazy agents: A new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 21937–21950. PMLR, 2023.
- [Liu *et al.*, 2023b] Yilin Liu, Guiyang Luo, Quan Yuan, Jinglin Li, Lei Jin, Bo Chen, and Rui Pan. Gplight: Grouped multi-agent reinforcement learning for large-scale traffic signal control. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 199–207, 2023.
- [Lowe *et al.*, 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390, 2017.
- [Oliehoek and Amato, 2016] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems. Springer, 2016.
- [Oliehoek *et al.*, 2008] Frans A Oliehoek, Matthijs TJ Spaan, Nikos Vlassis, and Shimon Whiteson. Exploiting locality of interaction in factored dec-pomdps. In *Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, 2008.
- [Pajarinen and Peltonen, 2011] Joni Pajarinen and Jaakko Peltonen. Efficient planning for factored infinite-horizon dec-pomdps. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 325, 2011.
- [Pignatelli *et al.*, 2023] Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, and

- Laura Toni. A survey of temporal credit assignment in deep reinforcement learning. *CoRR*, abs/2312.01072, 2023.
- [Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4292–4301. PMLR, 2018.
- [Salge *et al.*, 2014] Christoph Salge, Cornelius Glackin, and Daniel Polani. Changing the environment based on empowerment as intrinsic motivation. *Entropy*, 16(5):2789–2819, 2014.
- [Samvelyan *et al.*, 2019] Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2186–2188, 2019.
- [Schaul *et al.*, 2015] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.
- [Shannon and others, 1959] Claude E Shannon et al. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec.*, 4(142-163):1, 1959.
- [Son *et al.*, 2019] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Hostallero, and Yung Yi. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5887–5896. PMLR, 2019.
- [Strehl and Littman, 2008] Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for markov decision processes. *J. Comput. Syst. Sci.*, 74(8):1309–1331, 2008.
- [Sutton *et al.*, 2011] Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *10th International Conference on Autonomous Agents and Multiagent Systems*, pages 761–768, 2011.
- [Tang *et al.*, 2017] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2753–2762, 2017.
- [Wang *et al.*, 2020a] Tonghan Wang, Heng Dong, Victor R. Lesser, and Chongjie Zhang. ROMA: multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9876–9886. PMLR, 2020.
- [Wang *et al.*, 2020b] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. In *8th International Conference on Learning Representations*, 2020.
- [Wang *et al.*, 2021] Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. DOP: off-policy multi-agent decomposed policy gradients. In *9th International Conference on Learning Representations*, 2021.
- [Yang *et al.*, 2020] Jiachen Yang, Alireza Nakhaei, David Isele, Kikuo Fujimura, and Hongyuan Zha. CM3: cooperative multi-goal multi-stage multi-agent reinforcement learning. In *8th International Conference on Learning Representations*, 2020.
- [Yang *et al.*, 2024] Mingyu Yang, Yaodong Yang, Zhenbo Lu, Wengang Zhou, and Houqiang Li. Hierarchical multi-agent skill discovery. In *Advances in Neural Information Processing Systems* 36, 2024.
- [Yeh and Soo, 2024] Jhih-Ching Yeh and Von-Wun Soo. Toward socially friendly autonomous driving using multi-agent deep reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2573–2575, 2024.