Subspace-Aware Exploration for Sparse-Reward Multi-Agent Tasks

Pei Xu[1, 2], **Junge Zhang**[2], **Qiyue Yin**[2], **Chao Yu**[4], **Yaodong Yang**[5, 6], **Kaiqi Huang**[1,2,3]
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]CRISE, Institute of Automation, Chinese Academy of Sciences
[3]CAS, Center for Excellence in Brain Science and Intelligence Technology
[4]School of Computer Science and Engineering, Sun Yat-sen University
[5]Beijing Institute for General AI
[6]Institute for AI, Peking University
xupei2018@ia.ac.cn, {jgzhang,qyyin,kqhuang}@nlpr.ia.ac.cn
yuchao3@mail.sysu.edu.cn, yaodong.yang@pku.edu.cn

Abstract

Exploration under sparse rewards is a key challenge for multi-agent reinforcement learning problems. One possible solution to this issue is to exploit inherent task structures for an acceleration of exploration. In this paper, we present a novel exploration approach, which encodes a special structural prior on the reward function into exploration, for sparse-reward multi-agent tasks. Specifically, a novel entropic exploration objective which encodes the structural prior is proposed to accelerate the discovery of rewards. By maximizing the lower bound of this objective, we then propose an algorithm with moderate computational cost, which can be applied to practical tasks. Under the sparse-reward setting, we show that the proposed algorithm significantly outperforms the state-of-the-art algorithms in the multiple-particle environment, the Google Research Football and StarCraft II micromanagement tasks. To the best of our knowledge, on some hard tasks (such as `27m_vs_30m`) which have relatively larger number of agents and need non-trivial strategies to defeat enemies, our method is the first to learn winning strategies under the sparse-reward setting.

Introduction

Multi-agent reinforcement learning (MARL) is an increasingly important field. Many real-world problems (Swamy et al. 2020; Bazzan 2009) are naturally modelled using MARL technology. Recently, many works (Rashid et al. 2018; Lowe et al. 2017) have been proposed to address MARL problems. Although these works have made significant progress, they all focus on dense-reward multi-agent scenarios. However, in many real-world scenarios, rewards extrinsic to agents are extremely sparse (Pathak et al. 2017).

In this paper, we focus on exploration in sparse-reward multi-agent scenarios. Although classical exploration techniques such as count-based methods (Bellemare et al. 2016) perform well in single agent scenarios, recent studies (Mahajan et al. 2019; Liu et al. 2021) show that these techniques that strive to uniformly visit all states are no longer tractable in MARL. The reason is that the size of the state space grows exponentially with respect to the number of agents, which makes it extremely challenging to identify states worthy of being explored (Wang et al. 2019; Liu et al. 2021).

Previous work (Liu et al. 2021) has shown that exploiting structural prior on the reward function in multi-agent scenarios is a promising way to alleviate the exploration issue in MARL. In this paper, we focus on a special structural prior on the reward function in multi-agent tasks. The structural prior is that while the state space grows exponentially, the reward function typically depends on a small subset of the state space, i.e., sub-state space. This special structural prior is a common one, and widely exists in many multi-agent cooperation scenarios. For example, in real-world traffic light control scenario (Wei et al. 2018), as the number of traffic lights increases, the full-state space becomes very large, but the reward function only depends on the average waiting time of the vehicles, which is a small subset of the full-state space. In the academic field, there are also many such scenarios. For instance, in a multi-agent football task, the reward only relies on the position of the ball, or in a multi-agent fighting task, the reward is determined only by the health of enemies. Obviously, in these tasks, rewards are only associated with a subset of the full-state space, and exploring in this sub-state space enables to learn the whole task more efficiently.

In this work, we study how to effectively encode the structural prior into exploration, when specific domain knowledge is lacking. That is, we do not know which specific subspace the reward function depends on. While previous work has tackled the same problem (Liu et al. 2021), their method only works well for few tasks and follows a different exploration paradigm. We give a detailed comparison in Sec. .

Our basic idea is inspired by the principle of optimism in the face of uncertainty, which plays the central role in exploration methods in many fields (Strehl and Littman 2008). In single-agent RL, to identify states worthy of being explored, previous works (Bellemare et al. 2016; Burda et al. 2019b) assume that states with higher uncertainty are worthy of being explored and encourage agents to visit these states. Our approach extends this idea to sub-state space level. Since we do not know which specific sub-state space the reward function depends on, we assume that a sub-state space with higher uncertainty is more likely to be relevant to the reward function. Intuitively, an under-explored sub-state space has a higher uncertainty. Based on the idea, we propose the **S**ubspace-**A**ware **M**ulti-agent **E**xploration (SAME)

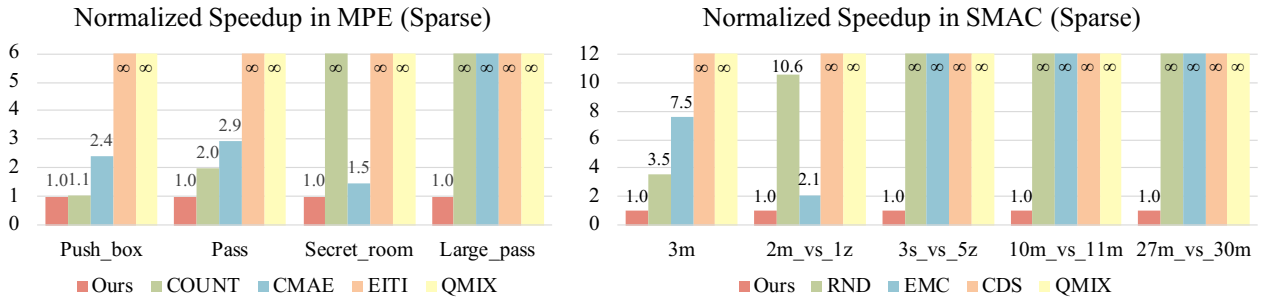

Figure 1: Normalized sample use by different methods with respect to SAME (smaller values are better). SAME consistently achieves a better sample efficiency compared to all other baselines. Infinity means that the method fails to achieve a success rate above 40% at a given step. Details for baselines are described in Sec. .

approach. SAME uses a novel entropic exploration objective in sub-state spaces to encourage agents to pay more attention to exploring sub-state spaces with higher uncertainty.

However, the proposed objective is sensitive to the number of agents, and needs to estimate state distribution induced by agents in high-dimensional sub-state spaces. These computational issues make it difficult to be applied to practical tasks. To this end, we propose an algorithm to encourage agents to improve the lower bound of the proposed objective. In this way, the computational cost with respect to the number of agents grows linearly rather than exponentially, and we only need to estimate the distribution in one-dimensional sub-state spaces which is relatively easy to achieve.

We evaluate SAME on three challenging environments: a discrete version of the multiple-particle environment (MPE) (Wang et al. 2019), the Google Research Football (GRF) (Kurach et al. 2020) and StarCraft II micromanagement (SMAC) (Samvelyan et al. 2019). In all experiments, we consider the sparse-reward setting. This means that agents get rewards only when they complete a given task. We show that SAME significantly outperforms the state-of-the-art baselines on almost all tasks (**RQ1** in Sec.). Fig 1 shows normalized sample size to achieve a success rate above 40% with respect to SAME. Moreover, to our best knowledge, on some tasks with relatively larger number of agents such as `27m_vs_30m`, SAME is the first to learn winning strategies under the sparse-reward setting. During the experiments, we observe (**RQ2** in Sec.) that SAME shows consistently strong performance on tasks with different number of agents, which confirms that our algorithm is relatively robust to the number of agents. We also observe (**RQ4** in Sec.) that compared to classical count-based exploration methods, SAME exhibits better coverage of states with rewards under the reward-free setting.

In summary, we make the following contributions: (i) We propose a new exploration approach which exploits the structural prior that the reward function typically depends on a small subset of the state space in multi-agent tasks; (ii) Based on our approach, we propose an algorithm that has moderate computational cost, and thus can be applied to practical tasks; and (iii) We show that our algorithm significantly outperforms the state-of-the-art algorithms on three challenging environments under the sparse-reward setting, including MPE, GRF and SMAC.

Preliminaries

Multi-Agent Markov Decision Process

A cooperative multi-agent system is modelled as a multi-agent Markov decision process (MDP). An n-agent MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{Z}, \mathcal{O}, n, \gamma, H)$. $\mathcal{S}$ is the full-state space of the environment. $\mathcal{A}$ is the action space of each agent. At each time step t, each agent's policy $\pi_i, i \in \mathcal{N} \equiv \{1, \dots, n\}$, selects an action $a_t^i \in \mathcal{A}$. All selected actions form a joint action $\boldsymbol{a}_t \in \mathcal{A}^n$. The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A}^n \to \Delta(\mathcal{S})$ maps the current full-state s_t and the joint action $\boldsymbol{a}_t$ to a distribution over the next full-state s_{t+1}. All agents receive a collective reward $r_t \in \mathbb{R}$, according to the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A}^n \to \mathbb{R}$. The objective of all agents' policies is to maximize the collective return $\sum_{t=0}^{H} \gamma^t r_t$, where $\gamma \in [0, 1]$ is the discount factor, H is the horizon, and r_t is the collective reward obtained at timestep t. Each agent i observes local observation $o_t^i \in \mathcal{Z}$, according to the observation function $\mathcal{O} : \mathcal{S} \times \mathcal{N} \to \mathcal{Z}$. All agents' local observations form a full observation $\boldsymbol{o}_t$. In this paper, we follow the standard centralized training with decentralized execution paradigm (CTDE) (Rashid et al. 2018).

Exploration from Sub-state Space

Formally, given an K-dimensional state space $\mathcal{S}$, the sub-state space $\mathcal{S}_\mathcal{K}$ associated with a set $\mathcal{K}$ is defined as:

$$\mathcal{S}_\mathcal{K} = \{proj_\mathcal{K}(s) : \forall s \in \mathcal{S}\}$$

where $proj_\mathcal{K}(s) = (s_e)_{e \in \mathcal{K}}$ restricts the space to elements e in set $\mathcal{K}$, i.e., $e \in \mathcal{K}$. Here, s_e is the e-th component of the full-state s, and $\mathcal{K}$ is a set from the power set of $\{1, \cdots, K\}$, i.e., $\mathcal{K} \in P(\{1, \cdots, K\})$, where P denotes the power set. For a state distribution d^π induced by a policy π (Hazan et al. 2019; Zhang et al. 2021a; Lee et al. 2019), the entropy of d^π over $\mathcal{S}_\mathcal{K}$ is denoted by $H(d^\pi, \mathcal{S}_\mathcal{K}) = -\mathbb{E}_{s \sim d^\pi} \log d^\pi(s, \mathcal{S}_\mathcal{K})$. We then use $\mathcal{E}_e$ to denote a one-dimensional sub-state space which only contains the e-th component of the full-state s, i.e., $|\mathcal{K}| = 1$.

Method

In this paper, we aim to alleviate the exploration issue in sparse-reward multi-agent tasks by exploiting the structural prior that the reward function typically depends on a small subset of the state space (i.e., sub-state space) in multi-agent tasks. In this section, we will discuss how to encode this prior into exploration so that agents learn the whole task more efficiently. An overview of the proposed approach is given in Fig. 2.

Exploration in Sub-state Spaces

One possible solution to encode the structural prior into exploration is to directly use classical exploration techniques in sub-state spaces. For example, if we use provably efficient maximum state entropy exploration methods (Hazan et al. 2019) in sub-state spaces, then we get the vanilla exploration objective

$$\mathcal{J}_{\text{van}}(\pi) = \sum_{\mathcal{K}\in P(\{1,\cdots,K\})} \mathcal{H}\left(d^{\pi}, \mathcal{S}_{\mathcal{K}}\right). \tag{1}$$

According to (Hazan et al. 2019; Zhang et al. 2021a), the bonus $b_{\text{sub}}^{\text{van}}(s_t, \boldsymbol{a}_t, s_{t+1})$ derived from Eq. 1 with respect to d^{π} is

$$b_{\text{sub}}^{\text{van}} = \sum_{\mathcal{K}\in P(\{1,\cdots,K\})} \left(-\log d^{\pi}\left(s_{t+1}, \mathcal{S}_{\mathcal{K}}\right) - 1\right). \tag{2}$$

Eq. 2 gives agents a higher bonus when agents visit a state that is less frequently visited in sub-state spaces. In this way, agents will be encouraged to keep exploring various sub-state spaces.

The main issue in Eq. 2 is that it gives the same weight to each sub-state space. This leads to agents having no bias for different sub-state spaces. Our aim is to find rewards quickly, rather than visiting all sub-state spaces uniformly. Therefore, a smarter approach is to make agents pay more attention to sub-state spaces that are relevant to rewards. However, accurately identifying which sub-state spaces are relevant to rewards requires domain knowledge, which is often difficult to obtain. In the absence of domain knowledge, inspired by the optimism principle in the face of uncertainty (Strehl and Littman 2008), we hypothesize that sub-state spaces with higher uncertainty (i.e., unfamiliar sub-state spaces) are more likely to be relevant to rewards. In other words, we give higher weights for sub-state spaces with higher uncertainty. In this paper, the uncertainty is defined as $U(d^{\pi}, \mathcal{S}_{\mathcal{K}}) = \frac{\log|\mathcal{S}_{\mathcal{K}}|}{H(d^{\pi}, \mathcal{S}_{\mathcal{K}})}$. Intuitively, a fully explored sub-state space has lower uncertainty. Formally, we use the new weighted objective to encourage exploration

$$\mathcal{J}_{\text{wt}}(\pi) = \sum_{\mathcal{K}\in P(\{1,\cdots,K\})} \log\left(1 + \frac{1}{U\left(d^{\pi}, \mathcal{S}_{\mathcal{K}}\right)}\right). \tag{3}$$

Similar to Eq. 1, agents also need to increase $H(d^{\pi}, \mathcal{S}_{\mathcal{K}})$ to maximize the exploration objective. However, in Eq. 3, increments of the entropy $H(d^{\pi}, \mathcal{S}_{\mathcal{K}})$ of different sub-state spaces have different effects on the objective. To show this more clearly, we derive the bonus $b_{\text{sub}}^{\text{wt}}(s_t, \boldsymbol{a}_t, s_{t+1})$ from Eq. 3 with respect to d^{π}

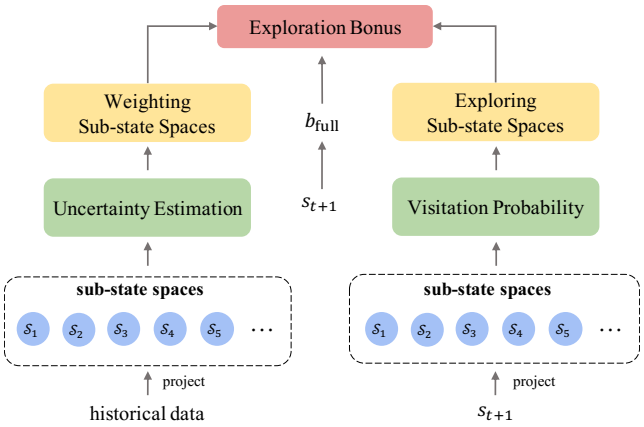

Figure 2: Overview of the proposed approach.

$$b_{\text{sub}}^{\text{wt}} = \sum_{\mathcal{K}\in P(\{1,\cdots,K\})} \underbrace{\frac{1}{\left(1 + \frac{1}{U(d^{\pi}, \mathcal{S}_{\mathcal{K}})}\right)\log|\mathcal{S}_{\mathcal{K}}|}}_{(1)} \times \underbrace{\left(-\log d^{\pi}\left(s_{t+1}, \mathcal{S}_{\mathcal{K}}\right) - 1\right)}_{(2)}. \tag{4}$$

The form of $b_{\text{sub}}^{\text{wt}}(s_t, \boldsymbol{a}_t, s_{t+1})$ clearly shows how it differs from Eq. 2. The first term weights sub-state spaces, and assigns higher values to sub-state spaces with higher uncertainty. This encourages agents to pay more attention to sub-state spaces with higher uncertainty. The second term is same as Eq. 2 and assigns higher values to states which are less frequently visited in the sub-state space level. The combination of these two terms achieves our goal. That is to make agents pay more attention to exploring sub-state spaces with higher uncertainty.

However, there are two computational issues in Eq. 4. The first issue is that $b_{\text{sub}}^{\text{wt}}$ is sensitive to the number of agents. For a task with N agents and each agent has M attributes, its full-state space has $N \times M$ dimensions, and all possible sub-state spaces are $2^{N\times M}$. This makes it impractical to consider all sub-state spaces when N is large. The second issue is that estimating $d^{\pi}(s_{t+1}, \mathcal{S}_{\mathcal{K}})$ in a high-dimensional sub-state space is non-trivial (Liu and Abbeel 2021). To alleviate the above computational issues, we consider a lower bound of the objective $\mathcal{J}_{\text{wt}}(\pi)$, which only considers one-dimensional sub-state spaces. Concretely, the lower bound is defined as (see Appendix for proof)

$$\widehat{\mathcal{J}}_{\text{wt}}(\pi) = \sum_{e=1}^{K} \log\left(1 + \frac{1}{U(d^{\pi}, \mathcal{E}_e)}\right) \leq \mathcal{J}_{\text{wt}}(\pi). \tag{5}$$

In this way, we only need to consider $N \times M$ sub-state spaces instead of $2^{N\times M}$ sub-state spaces. Moreover, since we only consider one-dimensional sub-state spaces, it is easy

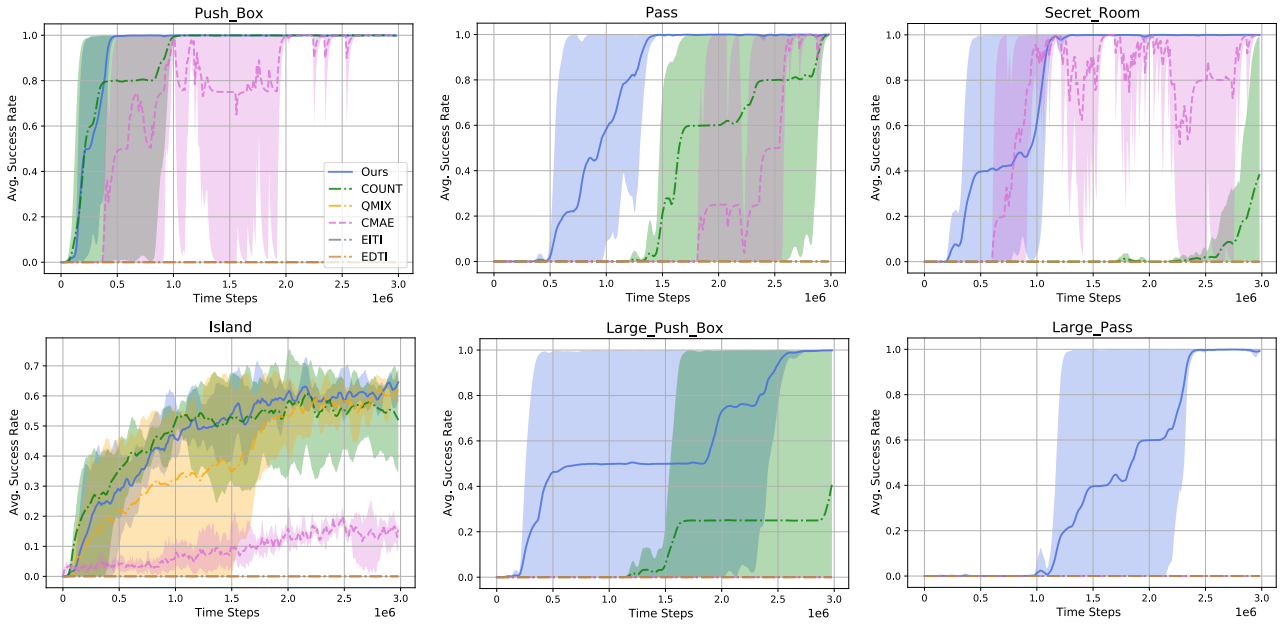

Figure 3: Comparison of our approach against baseline algorithms on MPE.

to estimate $d^\pi(s_{t+1}, \mathcal{E}_e)$. For example, we can discretize each dimension of the state space. Similar as Eq. 4, the bonus derived from the lower bound is

$$\hat{b}^{\text{wt}}_{\text{sub}} = \sum_{e=1}^{K} \frac{1}{\left(1 + \frac{1}{U(d^\pi, \mathcal{E}_e)}\right) \log|\mathcal{E}_e|} \left(-\log d^\pi(s_{t+1}, \mathcal{E}_e) - 1\right). \tag{6}$$

Combing with Full-state Space Bonus

Only relying on $\hat{b}^{\text{wt}}_{\text{sub}}$ to encourage exploration will make agents' behaviors lack guidance from the full-state space aspect, resulting in ignoring novel states in the full-state space level. In this paper, we use easy-to-implement count-based bonus as our $b_{\text{full}} = 1/\sqrt{N_{\text{tot}}(s_{t+1})}$, where N_{tot} stands for a state count. Now that we know how to calculate b_{full} and $\hat{b}^{\text{wt}}_{\text{sub}}$ respectively. The remaining problem is choosing an appropriate way to combine them. In high-level, we use the addition operation that is widely used in the literature (Zha et al. 2020; Wang et al. 2019) to combine b_{full} and $\hat{b}^{\text{wt}}_{\text{sub}}$. But, before performing the addition operation, b_{full} is also used as the coefficient of $\hat{b}^{\text{wt}}_{\text{sub}}$. This is to ensure the final bonus b_{tot} is asymptotically consistent (Zhang et al. 2021b), which means that the bonus will go to zero after enough exploration. Without asymptotically consistent, the optimal policy will be altered by the exploration bonus.

To discourage agents from visiting repeated states in an episode, we adopt local restrictions (Zhang et al. 2021b) for b_{full} and define a weakened local restriction for $\hat{b}^{\text{wt}}_{\text{sub}}$

$$m^{(e)}_{\text{sub}}(s_t, \boldsymbol{a}_t, s_{t+1}) = \mathbb{1}\big((s_{t+1})_e \neq (s_t)_e\big). \tag{7}$$

When the e-th dimension of state s_{t+1} is different from the e-th dimension of state s_t, $m^{(e)}_{\text{sub}}(s_t, \boldsymbol{a}_t, s_{t+1})$ is 1, otherwise it is 0. For b_{full}, we use the local restriction directly

$$m_{\text{full}}(s_t, \boldsymbol{a}_t, s_{t+1}) = \mathbb{1}\big(N_{\text{ep}}(s_{t+1}) = 0\big) \tag{8}$$

where N_{ep} stands for an episodic state count and is reset every episode. When the state s_{t+1} has not been visited before in this episode, $\mathbb{1}(N_{\text{ep}}(s_{t+1}) = 0)$ is 1, otherwise it is 0. Finally, the exploration bonus $b_{\text{tot}}(s_t, \boldsymbol{a}_t, s_{t+1})$ is

$$b_{\text{tot}} = \frac{m_{\text{full}}}{\sqrt{N_{\text{tot}}(s_{t+1})}} \left(1 + \beta \sum_{e=1}^{K} \frac{m^{(e)}_{\text{sub}}}{\left(1 + \frac{1}{U(d^\pi, \mathcal{E}_e)}\right) \log|\mathcal{E}_e|} \times \left(-\log d^\pi(s_{t+1}, \mathcal{E}_e) - 1\right)\right) \tag{9}$$

where β is a hyper-parameter. Our work guides exploration according to the exploration bonus b_{tot}. Therefore, agents receive an augmented reward which is the weighted sum of the task reward r and the exploration bonus b_{tot} at each timestep, $\hat{r} = r + w_1 b_{\text{tot}}$, where w_1 is the hyper-parameter to trade off exploration and exploitation.

Implementation

In this section, we discuss some important details about the implementation. To calculate $\hat{b}^{\text{wt}}_{\text{sub}}$, we need to estimate $|\mathcal{E}_e|$, $H(d^\pi, \mathcal{E}_e)$, and $d^\pi(s_{t+1}, \mathcal{E}_e)$. We use $\hat{\mathcal{E}}_e$ which is observed from data to approximate $\mathcal{E}_e$. For instance, if we observe $\mathcal{E}_e$ takes values $1, 2, 3$, then $|\hat{\mathcal{E}}_e| = 3$. We use empirical counts to estimate $d^\pi(s_{t+1}, \mathcal{E}_e)$ and $H(d^\pi, \mathcal{E}_e)$. To calculate $\hat{b}^{\text{wt}}_{\text{sub}}$ in the continuous state space (such as SMAC and GRF), we discretize each dimension of the state space into B equally spaced atomic states. Better state discretization techniques, such as hash-based counting (Tang et al. 2017), might lead to better performance, but we find that this simple approach

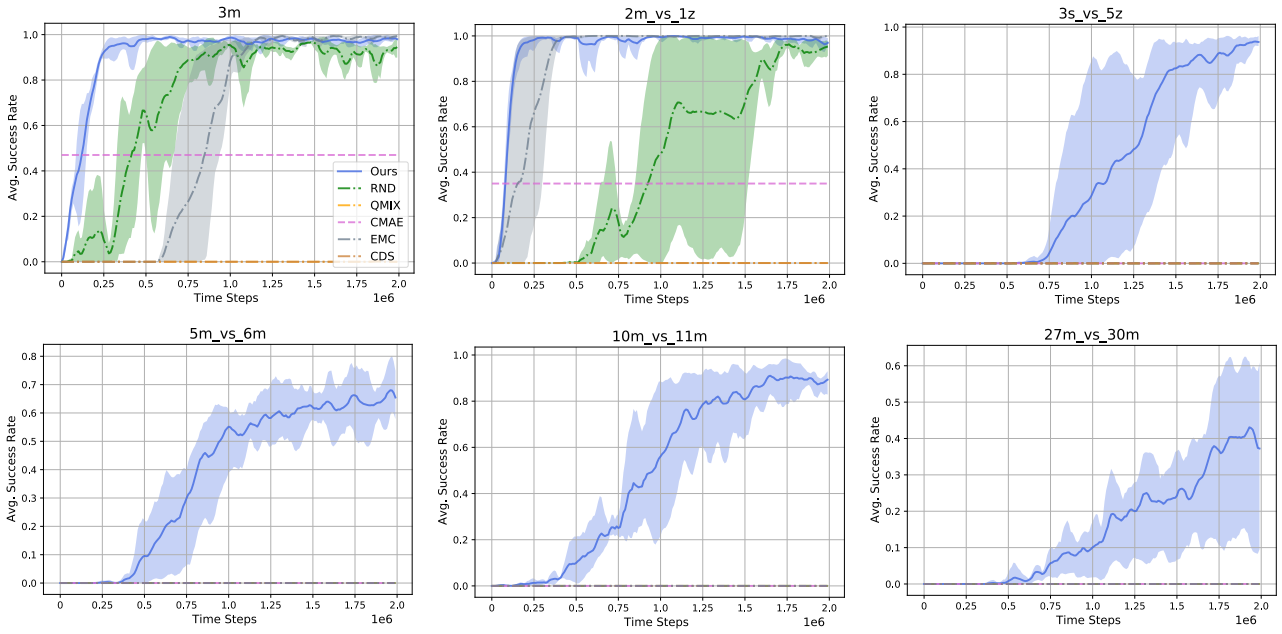

Figure 4: Results for the sparse-reward version of SMAC after training 2 million time steps. The proposed algorithm significantly outperforms all baseline algorithms.

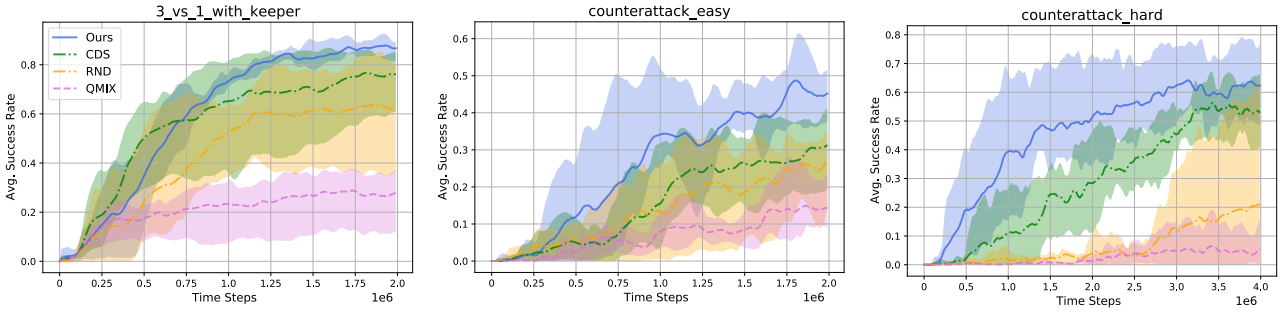

Figure 5: Comparison of our approach against baseline algorithms on Google Research Football.

is sufficient. In the continuous state space, we ignore m_{full} and use an alternate formula to calculate $m_{\text{sub}}^{(e)} = \|(s_{t+1})_e - (s_t)_e\|$. We use RND (Burda et al. 2019b) to calculate b_{full} in the continuous state space.

Experiments

The experiments are designed to answer the following research questions: **RQ1**: how is SAME compared with the state-of-the-art exploration algorithms on benchmark multi-agent environments in the sparse-reward setting (Sec)? **RQ2**: How will SAME perform in tasks with more agents (Sec)? **RQ3**: how important each component is in SAME (Sec)? **RQ4**: how will SAME explore environments (Sec)? All experiments run with five random seeds. Details for environments and training are given in Appendix.

Experiments on Standard Multi-agent Tasks

To study **RQ1**, we evaluate SAME on three challenging environments: (1) a discrete version of the multiple-particle environment (MPE) (Liu et al. 2021); (2) the StarCraft II micromanagement (SMAC) (Samvelyan et al. 2019); and (3) the Google Research Football (GRF) (Kurach et al. 2020). In all environments, we consider the sparse-reward setting.

Experimental Setup. In MPE, following previous works (Wang et al. 2019; Liu et al. 2021), we consider four standard tasks: `Push_Box`, `Pass`, `Secret_Room` and `Island`. We also consider more challenging tasks `Large_Push_Box` and `Large_Pass`. To evaluate our algorithm on environments with continuous state space, we consider six standard tasks in SMAC: `3m`, `2m_vs_1z`, `3s_vs_5z`, `5m_vs_6m`, `10m_vs_11m` and `27m_vs_30m`. We consider the sparse reward setting, which means agents see a reward of $+1$ only when all enemies are taken care of. In GRF (Kurach et al. 2020), following pre-

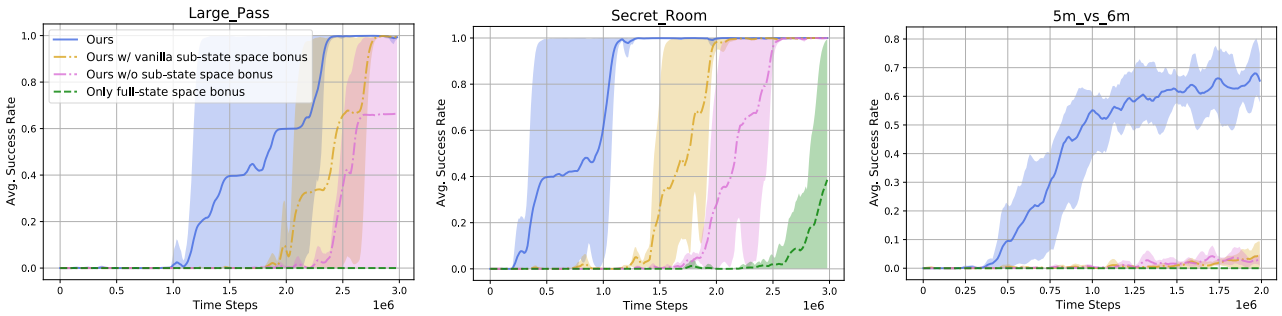

Figure 6: Ablation study for SAME. Without sub-state space level bonus, the performance drops significantly in all tasks. This confirms the importance of our sub-state space bonus.

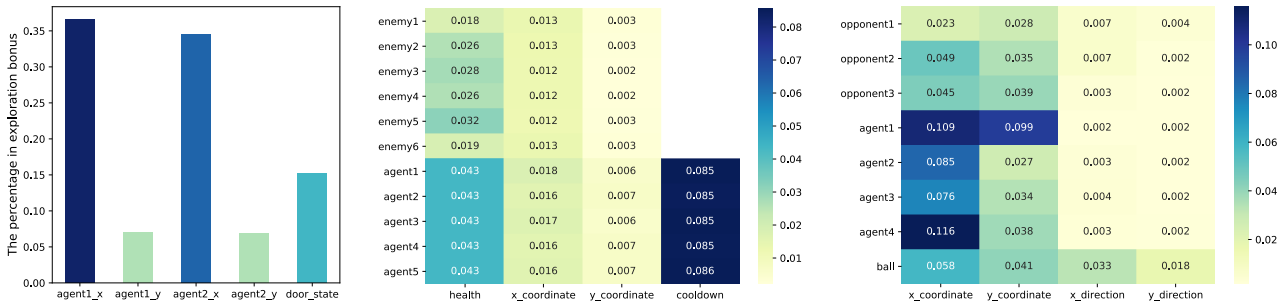

Figure 7: We show the percentage of each one-dimensional sub-state space's bonus in the overall exploration bonus in `Secret_Room` (left), `5m_vs_6m` (middle) and `counterattack_hard` (right). Larger values mean more attention. Each cell in last two figures represents a one-dimensional sub-state space, such as the health of agent 1. The values of all cells in a figure add up to one.

vious work (Chenghao et al. 2021), we consider three tasks: `3_vs_1_with_keeper`, `counterattack_easy` and `counterattack_hard`. In GRF tasks, only scoring leads to rewards.

Baselines. We consider several baselines: **QMIX** (Rashid et al. 2018) is a popular value-based method for MARL; **COUNT** includes a count-based bonus on top of QMIX; CMAE (Liu et al. 2021) learns an exploration policy by selecting goals from many restricted spaces; **EITI** and **EDTI** (Wang et al. 2019) capture the influence of one agent's behaviors on others; **EMC** (Zheng et al. 2021) uses prediction errors of individual Q-values as intrinsic rewards for coordinated exploration; **CDS** (Chenghao et al. 2021) introduces diversity in both optimization and representation to encourage extensive exploration. In SMAC and GRF, the count-based bonus and our b_{full} are approximated by **RND** (Burda et al. 2019b).

Results on MPE. We first compare SAME with baselines on MPE tasks. The training curves are included in Fig. 3. The results of CMAE are obtained using the publicly available code released by the authors. EITI and EDTI, which need to learn dynamics, both fail in all tasks. As we expected, COUNT which combines with a count-based exploration bonus can solve easy tasks, but does not perform well on hard tasks. On simple tasks, CMAE can learn winning strategies but the performance is unstable, while on difficult tasks CAME fails. We think this is mainly because CMAE ignores the full-state information and its multi-stage paradigm. Our algorithm shows amazing sample efficiency on all tasks. Specifically, only SAME is able to solve `Large_Pass`. On `Pass` and `Large_Push_Box`, SAME explores faster than all baselines.

Results on SMAC. Next, we evaluate our algorithm in more challenging tasks with continuous state space. The training curves are included in Fig. 4. Since CMAE does not provide an implementation on SMAC, we get the results of CAME from the original paper (Liu et al. 2021). As shown in Fig. 4, QMIX which relies on random exploration and CDS which introduces diversity do not learn a winning strategy in all tasks. RND and EMC can solve tasks with fewer agents, such as `3m` and `2m_vs_1z` tasks. However, as we expected, they both fail in tasks with more agents. In contrast, our SAME works well in all tasks. Concretely, our algorithm is more efficient than all baselines on tasks with fewer agents. More importantly, our algorithm also achieves strong performance when the number of agents increases, such as `3s_vs_5z` and `5m_vs_6m`.

Results on GRF. Next, we evaluate our algorithm on three challenging Google Research Football (GRF) offensive scenarios: `3_vs_1_with_keeper`, `counterattack_easy` and `counterattack_hard`. In GRF, all experiments follow the training settings of

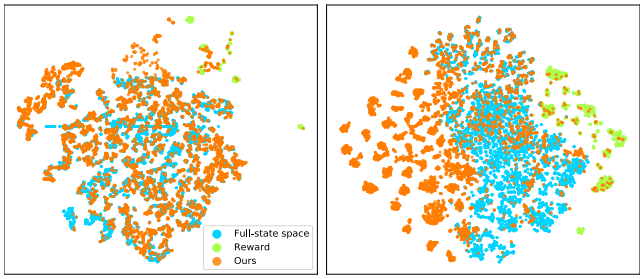

Figure 8: t-SNE plot of states randomly sampled from the 1M steps training without the extrinsic reward in `Secret_Room` (left) and `5m_vs_6m` (right). States with rewards are collected by other agents are trained with the extrinsic reward.

CDS (Chenghao et al. 2021), except that all experiments use TD(λ) to speed up training. The training curves are reported in Fig. 5. We observe that, as the difficulty of the task increases, the advantages of our algorithm become more obvious. On easy tasks, our algorithm slightly outperforms the current state-of-the-art algorithm CDS, but on harder tasks, our algorithm significantly outperforms all baselines.

More Agents. To study **RQ2**, we consider tasks with more agents. Concretely, we consider `10m_vs_11m` with 21 agents, and `27m_vs_30m` with 57 agents. Obviously, CMAE (Liu et al. 2021) is unable to be applied to these tasks because the number of agents is so large. Since our algorithm only needs to consider K one-dimensional sub-state spaces, it can be easily applied to these complex tasks. As shown in Fig. 4, SAME can still learn winning strategies. To our knowledge, SAME is the first to learn winning strategies in these tasks under the sparse-reward setting.

Discussion. Through extensive experiments on three challenging standard multi-agent environments, we make two observations. First, the results support the view of previous work (Mahajan et al. 2019; Liu et al. 2021) that conventional exploration strategies in single agent scenarios, such as COUNT and RND, are no longer tractable in multi-agent scenarios. Second, our algorithm demonstrates superior performance in all tasks, confirming that it is feasible to accelerate the discovery of rewards by encoding the structural prior on the reward function into exploration.

Ablations and Analysis

To study **RQ3**, we perform ablation studies on our algorithm. To confirm the effectiveness of the sub-state space bonus, we first report the performance by setting the sub-state space bonus to a constant 1 ('Ours w/o sub-state space bonus'). As shown in Fig. 6, without the sub-state space bonus, the performance drops significantly. This confirms the importance of our sub-state space bonus. Then, we verify if our bonus $b_{\text{sub}}^{\text{wt}}$ in Eq. 6 is better than $b_{\text{sub}}^{\text{van}}$. As we expected, the performance trained with $b_{\text{sub}}^{\text{van}}$ ('Ours w/ vanilla sub-state space bonus') drops significantly. This confirms the importance of the first term in Eq. 6. Last, by comparing the performance with only full-state space bonus, and the performance with full-state space bonus and local restrictions ('Ours w/o sub-state space bonus'), we observe that local restrictions are beneficial for exploration.

Exploration Behavior Analysis. To study **RQ4**, we report the percentage of each one-dimensional sub-state space's bonus in the overall exploration bonus during 500K steps training in Fig 7. Larger values mean more attention. That is, exploration behaviors of agents are mainly guided by bonuses which are from sub-state spaces with high values. As we expected, our algorithm has a clear preference for different sub-state spaces. For example, our algorithm pays more attention to x-coordinates of agents and states of doors in `Secret_Room`. In this task, agents need to enter the correct room by switching states of doors. Therefore, paying more attention to the states of doors helps to solve the task. In `5m_vs_6m`, our algorithm focuses on the weapon cooldown of agents, agents' health and enemies' health. In this task, the goal is to take care of all enemies. Thence, focusing more on these sub-state spaces obviously helps agents to find rewards faster. Overall, sub-state spaces that our algorithm focuses on can help agents to find rewards. This is consistent with our motivation.

To better understand exploration behaviors of our agents, we use t-SNE (Van der Maaten and Hinton 2008) to visualize some visited states during 1M steps training without the extrinsic reward. As shown in Fig. 8, we plot states visited by our algorithm (orange), states visited by classical full-state space exploration (blue). We also plot states with the extrinsic reward (green) that are collected by other agents, which are trained with the extrinsic reward. We observe that some states visited by our algorithm overlap with rewarded states in the reward-free setting. This indicates our algorithm better covers states with rewards. This is expected because our algorithm encodes a structural prior on the reward function into exploration, whereas the full-state space exploration only considers visitation counts.

Related Work

Many exploration techniques have been studied for single-agent deep reinforcement learning problems. Among them, two types of intrinsic reward methods are the most popular. One type is count-based methods which encourage agents to visit novel states (Strehl and Littman 2008; Bellemare et al. 2016). The other class of methods rely on prediction errors for problems related to the agent's transitions (Pathak et al. 2017; Burda et al. 2019a,b; Badia et al. 2020). However, directly applying these methods in MARL is impractical due to large scale of exploration space.

Recently, exploration approaches designed for multi-agent scenarios have been proposed. EITI and EDTI (Wang et al. 2019) capture the influence of one agent's behaviors on others, and agents are encouraged to visit states that will change other agents' behaviors. More recently, EMC (Zheng et al. 2021) uses prediction errors of individual Q-values as intrinsic rewards for coordinated exploration. CDS (Chenghao et al. 2021) maximizes the mutual information between agents' identities and their trajectories to encourage extensive exploration and diverse individualized behaviors. However, these methods ignore the structure of the reward func-

tion, resulting in inefficient explorations in complex environments under the sparse-reward setting.

The most relevant work to our approach is CMAE (Liu et al. 2021). Our approach differs from CMAE in the following two aspects. First, CMAE is a multi-stage method that divides the whole training process into multiple stages according to heuristic rules and has to retrain a new exploration policy from scratch at each stage, which leads to inefficient exploration. In contrast, our approach follows the bonus-based exploration paradigm and can be seamlessly combined with existing methods (Rashid et al. 2018; Chenghao et al. 2021). Second, CMAE is sensitive to the number of agents. The reason is that CMAE needs to explicitly select a sub-state space to be explored, resulting in the need to enumerate a large number of sub-state spaces. This makes CAME difficult to be applied to tasks with larger number of agents. In contrast, our algorithm has moderate computational cost, and is insensitive to the number of agents.

Conclusion

In this paper, we aim to alleviate the exploration issue in sparse-reward MARL tasks by exploiting the structural prior that the reward function typically depends on a small subset of the state space. To encode the structural prior into exploration, we propose a new exploration objective in sub-state spaces. Moreover, based on the objective, we propose an algorithm with moderate computational cost, which can be applied to practical tasks. We evaluate our algorithm on three challenging exploration environments under the sparse-reward setting. Results show that our algorithm pushes forward state-of-the-art. Moreover, on some hard tasks, our algorithm can still learn winning strategies, while other algorithms fail. One limitation of the current work is that it is based on the CTDE framework, which assumes that the global state is available at training. Although CTDE is currently the most popular framework, the global state is difficult to obtain in real-world scenarios. In addition, in many scenarios the information of other agents is not available due to privacy issues. These problems have inspired recent studies about decentralized training. In the future, we hope to extend our core idea to the decentralized training setting.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (Grant No. 61721004 and No.61876181), the Projects of Chinese Academy of Science (Grant No. QYZDB-SSW-JSC006), the Youth Innovation Promotion Association CAS, and the Beijing Nova Program of Science and Technology (Grand No. Z191100001119043).

References

Badia, A. P.; Sprechmann, P.; Vitvitskyi, A.; Guo, D.; Piot, B.; Kapturowski, S.; Tieleman, O.; Arjovsky, M.; Pritzel, A.; Bolt, A.; et al. 2020. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*.

Bazzan, A. L. 2009. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(3): 342–375.

Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 1471–1479.

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; and Efros, A. A. 2019a. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019b. Exploration by random network distillation. In *International Conference on Learning Representations*.

Chenghao, L.; Wang, T.; Wu, C.; Zhao, Q.; Yang, J.; and Zhang, C. 2021. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34.

Hazan, E.; Kakade, S.; Singh, K.; and Van Soest, A. 2019. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, 2681–2691. PMLR.

Kurach, K.; Raichuk, A.; Stanczyk, P.; Zajac, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4501–4510.

Lee, L.; Eysenbach, B.; Parisotto, E.; Xing, E.; Levine, S.; and Salakhutdinov, R. 2019. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*.

Liu, H.; and Abbeel, P. 2021. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34: 18459–18473.

Liu, I.-J.; Jain, U.; Yeh, R. A.; and Schwing, A. 2021. Cooperative Exploration for Multi-Agent Deep Reinforcement Learning. In *International Conference on Machine Learning*, 6826–6836. PMLR.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*.

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

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

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

Samvelyan, M.; Rashid, T.; Schroeder de Witt, C.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.;

Foerster, J.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2186–2188.

Strehl, A. L.; and Littman, M. L. 2008. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8): 1309–1331.

Swamy, G.; Reddy, S.; Levine, S.; and Dragan, A. D. 2020. Scaled autonomy: Enabling human operators to control robot fleets. In *2020 IEEE International Conference on Robotics and Automation*, 5942–5948. IEEE.

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Chen, O. X.; Duan, Y.; Schulman, J.; DeTurck, F.; and Abbeel, P. 2017. # Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, 2753–2762.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, T.; Wang, J.; Wu, Y.; and Zhang, C. 2019. Influence-based multi-agent exploration. In *International Conference on Learning Representation*.

Wei, H.; Zheng, G.; Yao, H.; and Li, Z. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2496–2505.

Zha, D.; Ma, W.; Yuan, L.; Hu, X.; and Liu, J. 2020. Rank the Episodes: A Simple Approach for Exploration in Procedurally-Generated Environments. In *International Conference on Learning Representations*.

Zhang, T.; Rashidinejad, P.; Jiao, J.; Tian, Y.; Gonzalez, J. E.; and Russell, S. 2021a. Made: Exploration via maximizing deviation from explored regions. *Advances in Neural Information Processing Systems*, 34.

Zhang, T.; Xu, H.; Wang, X.; Wu, Y.; Keutzer, K.; Gonzalez, J. E.; and Tian, Y. 2021b. NovelD: A Simple yet Effective Exploration Criterion. *Advances in Neural Information Processing Systems*, 34.

Zheng, L.; Chen, J.; Wang, J.; He, J.; Hu, Y.; Chen, Y.; Fan, C.; Gao, Y.; and Zhang, C. 2021. Episodic Multi-agent Reinforcement Learning with Curiosity-driven Exploration. *Advances in Neural Information Processing Systems*, 34.