Exploration via Joint Policy Diversity for Sparse-Reward Multi-Agent Tasks

Pei Xu[1,2] , **Junge Zhang**[2] , **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
xupei2018@ia.ac.cn, {jgzhang,kqhuang}@nlpr.ia.ac.cn

Abstract

Exploration under sparse rewards is a key challenge for multi-agent reinforcement learning problems. Previous works argue that complex dynamics between agents and the huge exploration space in MARL scenarios amplify the vulnerability of classical count-based exploration methods when combined with agents parameterized by neural networks, resulting in inefficient exploration. In this paper, we show that introducing constrained joint policy diversity into a classical count-based method can significantly improve exploration when agents are parameterized by neural networks. Specifically, we propose a joint policy diversity to measure the difference between current joint policy and previous joint policies, and then use a filtering-based exploration constraint to further refine this joint policy diversity. Under the sparse-reward setting, we show that the proposed method significantly outperforms the state-of-the-art methods in the multiple-particle environment, the Google Research Football, and StarCraft II micromanagement tasks. To the best of our knowledge, on the hard `3s_vs_5z` task which needs non-trivial strategies to defeat enemies, our method is the first to learn winning strategies without domain knowledge under the sparse-reward setting.

1 Introduction

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

To enable agents to handle these sparse-reward scenarios well, studies on how to improve the ability of agents to explore environments are essential. Many great works [Jin *et al.*, 2018; Hazan *et al.*, 2019; Xu *et al.*, 2021] have been proposed in RL. However, recent studies [Liu *et al.*, 2021; Ryu *et al.*, 2022; Mahajan *et al.*, 2019] experimentally show that classical exploration methods, such as count-based exploration [Strehl and Littman, 2008; Jin *et al.*, 2018] and variants [Burda *et al.*, 2018b; Bellemare *et al.*, 2016] which extend these methods to high-dimensional state spaces, do not work well in MARL scenarios when agents are parameterized by neural networks. They believe this is caused by complex dynamics between agents and the huge exploration space in MARL scenarios that amplify the vulnerability of these classical methods when encountering neural networks.

To handle the issue, some exploration methods [Wang *et al.*, 2019; Zheng *et al.*, 2021; Chenghao *et al.*, 2021] for multi-agent tasks have been proposed. For example, [Wang *et al.*, 2019] proposes an exploration bonus by preferring states that can affect transitions. [Zheng *et al.*, 2021] propose a better estimate of the uncertainty of the state for MARL to calculate bonuses. Overall, most of these methods enhance exploration by making better use of state-level information. However, relying only on exploration bonuses based on state-level information is not sufficient to achieve efficient exploration [Rashid *et al.*, 2019].

To this end, this work studies how other information can be used to encourage exploration. We note that our goal is to make agents to learn policies that can solve a given task. Motivated by this, a natural question is *whether policy-level information is helpful for exploration*. In this paper, we focus on using policy-level information (i.e., joint policy diversity), to enhance classical exploration methods based on state-level information, such as count-based exploration. Specifically, we force agents to choose different behaviors compared to the previous episodes by maximizing the difference between the current joint policy and previous joint policies. The difference is measured by the proposed *unbalanced diversity measurement*. It is important to highlight that unconstrained optimization of the differences between joint policies does not necessarily improve exploration. Due to the huge joint policy space, this is likely to encourage agents to learn a different, yet unhelpful joint policy for exploration (e.g., always staying in a state). To this end, we propose a *filtering-based exploration constraint* to force agents to optimize their joint policy in the direction that improves exploration.

We empirically evaluate the proposed method on three challenging environments: a discrete version of the multiple-

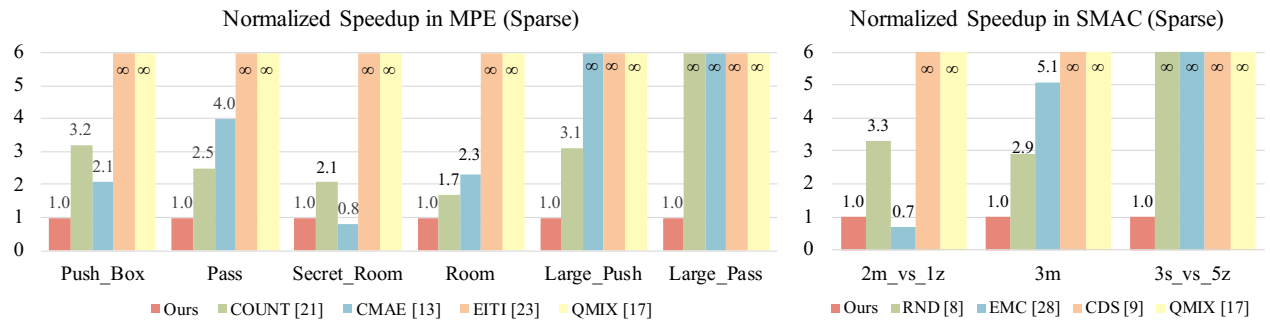

Figure 1: Normalized sample use by different methods with respect to our method (smaller values are better). Our method achieves a better or comparable sample efficiency compared to baselines. Infinity means that the method fails to achieve a success rate above 50% at a given step.

particle environment (MPE) [Wang *et al.*, 2019], the Google Research Football [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 our method significantly outperforms the state-of-the-art baselines on almost all tasks (**RQ1** in Sec. 5.1). Fig. 1 shows a normalized sample size to achieve a success rate above 50% with respect to our method. Moreover, to our best knowledge, on some hard tasks such `3s_vs_5z`, our method is the first to learn winning strategies without domain knowledge under the sparse-reward setting. To better understand the exploration behaviors of the proposed method, we present extensive experiments in the reward-free setting. Results show that our method, which is on top of a classical bonus-based method (i.e., count-based method), can explore significantly more states compared to the classical method (**RQ2** in Sec. 5.2).

In summary, we make three contributions: (i) We propose a novel method to enhance classical exploration methods based on state-level information by using policy-level diversity for sparse-reward multi-agent tasks; (ii) We propose a constrained joint policy diversity to measure the difference between current joint policy and previous joint policies; and (iii) Our method significantly outperforms the state-of-the-art methods on three challenging benchmarks under the sparse-reward setting, including MPE, GRF, and SMAC.

2 Related Work

Many exploration techniques have been studied for single-agent deep reinforcement learning problems. Among them, two types of bonus-based 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 relies on prediction errors for problems related to the agent's transitions [Pathak *et al.*, 2017; Burda *et al.*, 2018a; Burda *et al.*, 2018b; Badia *et al.*, 2020]. However, these methods focus on how to better estimate the state uncertainty in the high-dimensional state space. Recent studies [Mahajan *et al.*, 2019; Wang *et al.*, 2019; Liu *et al.*, 2021] experimentally show that these methods do not work well in MARL scenarios which have larger exploration spaces and complex dynamics between agents.

Recently, some exploration methods 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 exploration. CDS [Chenghao *et al.*, 2021] maximizes the mutual information between agents' identities and their trajectories to encourage extensive exploration. In summary, these methods introduce some heuristic principles to make better use of state-level information. In contrast, our work focuses on policy-level information.

More recently, some researchers have turned to other technical lines to improve exploration in MARL scenarios. CMAE [Liu *et al.*, 2021] proposes a goal-conditioned exploration method that can cleverly use domain knowledge. [Zhang *et al.*, 2021] uses a learned centralized environmental model to replace the real environment, thus improving sample efficiency. [Ryu *et al.*, 2022] uses a powerful simulator to modify the distribution of initial states, making agents more likely to find rewards. However, all of these methods have some insurmountable issues, such as the difficulty of obtaining domain knowledge and the accuracy of the learned model. In contrast, our work is a bonus-based method that does not depend on domain knowledge or a learned model.

3 Preliminaries

A cooperative multi-agent task is modeled as a multi-agent Markov decision process (MDP). An n-agent MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}, \mathcal{R}, \mathcal{Z}, \mathcal{O}, n, \gamma, H)$. $\mathcal{S}$ is the state space. $\mathcal{A}$ 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 $\mathbb{P} : \mathcal{S} \times \mathcal{A}^n \to \Delta(\mathcal{S})$ maps the current state s_t and the joint action $\boldsymbol{a}_t$ to a distribution over the next 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 $\boldsymbol{o}_t$. All agent's policy π^i form a joint policy $\boldsymbol{\pi} : \mathcal{O}^n \to \Delta(\mathcal{A}^n)$. In this paper, we follow the standard centralized training with decentralized execution paradigm (CTDE) [Rashid *et al.*, 2018].

4 Method

In this section, we present a novel exploration method for sparse-reward multi-agent tasks. The proposed method exploits information from the joint policy level to improve classical count-based methods [Strehl and Littman, 2008; Burda *et al.*, 2018b], which are based on state uncertainty and do not work well with neural networks. An overview of the proposed method is given in Fig. 2.

4.1 Joint Policy Diversity

One possible solution to use information from the joint policy level to drive exploration is to encourage diversity between joint policies. Specifically, in the k-th parameter update, we optimize the current joint policy $\boldsymbol{\pi}_\theta$ with the constraint that $\boldsymbol{\pi}_\theta$ is distinct from previous joint policies (i.e., $\boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_{k-1}$). To implement this idea, a natural choice is using KL-divergence to measure the distance between joint policies. Formally, the distance is defined by

$$\mathcal{J}_{\text{kl}}(\boldsymbol{\pi}_\theta) = \sum_{j=1}^{k-1} \frac{1}{k-1} \mathcal{D}_{\text{kl}}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_j). \tag{1}$$

However, there are several issues in Eq. 1. The first issue is that KL-divergence is not suitable in our setting. The accumulative KL-divergence of a trajectory is defined as

$$\begin{aligned} \mathcal{D}_{\text{kl}}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_j) &= \mathbb{E}_{\tau\sim\boldsymbol{\pi}_\theta}\Big[\sum_t \log \frac{\boldsymbol{\pi}_\theta(\boldsymbol{a}_t|\boldsymbol{o}_t)}{\boldsymbol{\pi}_j(\boldsymbol{a}_t|\boldsymbol{o}_t)}\Big] \\ &= \mathcal{H}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_j) - \mathcal{H}(\boldsymbol{\pi}_\theta) \end{aligned} \tag{2}$$

where $\mathcal{H}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_j)$ is the cross-entropy between $\boldsymbol{\pi}_\theta$ and $\boldsymbol{\pi}_j$, and $\mathcal{H}(\boldsymbol{\pi}_\theta)$ is the entropy of $\boldsymbol{\pi}_\theta$. As shown in the above derivation, using KL-divergence as the distance measure would inherently encourage learning a joint policy with small entropy $\mathcal{H}(\pi_\theta)$, which is undesirable in our problems since a low entropy joint policy is harmful for exploration. Thus, we ignore $\mathcal{H}(\boldsymbol{\pi}_\theta)$ in Eq. 2. In other words, we choose to use *Cross-Entropy* instead of KL-divergence. Formally, the distance based on the cross-entropy is defined as

$$\begin{aligned} \mathcal{J}_{\text{ce}}(\boldsymbol{\pi}_\theta) &= \sum_{j=1}^{k-1} \frac{1}{k-1} \mathcal{H}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_j) \\ &= \sum_{j=1}^{k-1} \frac{1}{k-1} \mathbb{E}_{\tau\sim\boldsymbol{\pi}_\theta}\Big[\sum_t \log \frac{1}{\boldsymbol{\pi}_j(\boldsymbol{a}_t|\boldsymbol{o}_t)}\Big]. \end{aligned} \tag{3}$$

The second issue is that, to calculate Eq. 3, we need to calculate the distance between the current joint policy and the previous $k-1$ joint policies at the k-th parameter update. As the training process continues, we need to consider a large number of previous joint policies, which leads to an unacceptable consumption of computational resources. Consequently, we introduce a joint policy mixture $\boldsymbol{\pi}_{\text{mix}}$ which is the average of all previous joint policies. Formally, in the k-th parameter update, the joint policy mixture $\boldsymbol{\pi}_{\text{mix}}$ is defined as

$$\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}|\boldsymbol{o}) = \sum_{j=1}^{k-1} \frac{1}{k-1} \boldsymbol{\pi}_j(\boldsymbol{a}|\boldsymbol{o}) \tag{4}$$

Then, according to Jensen's inequality, we have

$$\widehat{\mathcal{J}}_{\text{ce}}(\boldsymbol{\pi}_\theta) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}_\theta}\Big[\sum_t \log \frac{1}{\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)}\Big] \le \mathcal{J}_{\text{ce}}(\boldsymbol{\pi}_\theta). \tag{5}$$

The above inequality tells us that the cross-entropy between the current joint policy $\boldsymbol{\pi}_\theta$ and the joint policy mixture $\boldsymbol{\pi}_{\text{mix}}$ is a lower bound of the original optimization objective $\mathcal{J}_{\text{ce}}(\boldsymbol{\pi}_\theta)$. Thus, we can optimize the original objective by maximizing this lower bound which has a moderate computational cost. In this paper, we get a joint policy mixture $\boldsymbol{\pi}_{\text{mix}}$ by using past joint action/observation histories to train networks. Details of implementation are given in supplementary material.

Next, we discuss how to further modify Eq. 5. In our setting, agents are expected to choose actions that were less frequently chosen before, which is helpful for exploration. Therefore, we propose an *Unbalanced Cross-Entropy* by adding a penalty term into the original cross-entropy. For an observation-action pair $(\boldsymbol{o}_t, \boldsymbol{a}_t)$ from a joint policy $\boldsymbol{\pi}_\theta$ and a policy mixture $\boldsymbol{\pi}_{\text{mix}}$, when the probability $\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)$ is higher than a fully random joint policy $\boldsymbol{\pi}_{\text{rand}}$ (i.e., selecting each joint action with the same probability), the penalty term will reduce the contribution of the observation-action $(\boldsymbol{o}_t, \boldsymbol{a}_t)$ pair to the distance. Formally, the unbalanced cross-entropy is defined as

$$\begin{aligned} \mathcal{H}_{\text{uce}}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_{\text{mix}}) = \mathbb{E}_{\tau\sim\pi_\theta}\Bigg[&\sum_t \log \frac{1}{\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)} \\ &+ \sum_t \mathbb{I}\Big[\frac{\boldsymbol{\pi}_{\text{rand}}(\boldsymbol{a}_t|\boldsymbol{o}_t)}{\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)} \le 1\Big] \log \beta\Bigg] \end{aligned} \tag{6}$$

where $\mathbb{I}[\cdot]$ denotes an indicator function and $\beta \in (0, 1]$ is a penalty factor. The unbalanced cross-entropy $\mathcal{H}_{\text{uce}}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_{\text{mix}})$

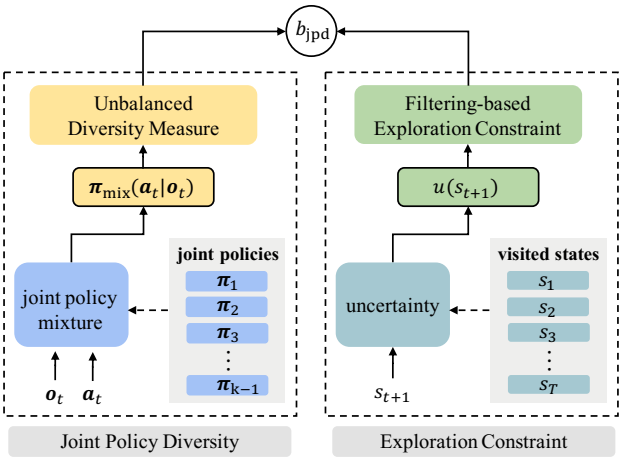

Figure 2: Overview of the proposed method.

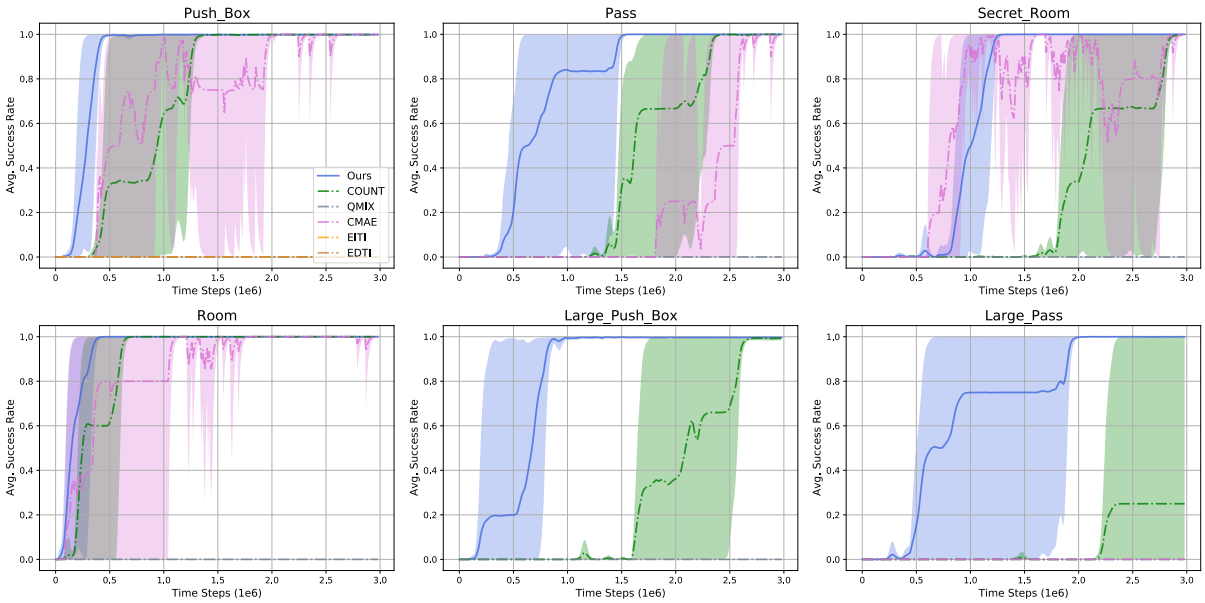

Figure 3: Comparison of our method against baseline methods on MPE. The proposed method significantly outperforms all baseline methods.

recovers the original cross-entropy $\mathcal{H}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}_{\text{mix}})$, when β is set to 1.

To calculate the distance in Eq. 6, we need to estimate the joint action probability. To this end, we use an auto-regressive form [Fu *et al.*, 2022] to represent a joint policy mixture $\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}|\boldsymbol{o})$. Given an execution order $X = \{x_1, \cdots, x_n\}$, we can factorize a joint policy $\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}|\boldsymbol{o})$ into the form of

$$\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}|\boldsymbol{o}) \approx \boldsymbol{\pi}^{\text{ar}}_{\text{mix}}(\boldsymbol{a}|\boldsymbol{o}) = \prod_{i=1}^{n} \pi^{x_i}_{\theta^{x_i}}(a^{x_i}|o^{x_i}, a^{x_1}, \cdots, a^{x_{i-1}}). \tag{7}$$

where $\pi^{x_i}_{\theta^{x_i}}$ is the x_i-th agent. In this way, we can obtain an estimate of the joint action probability without directly learning a joint policy.

In summary, we use the joint policy diversity

$$\widehat{\mathcal{J}}^{\text{ar}}_{\text{uce}}(\boldsymbol{\pi}_\theta) = \mathcal{H}_{\text{uce}}(\boldsymbol{\pi}_\theta, \boldsymbol{\pi}^{\text{ar}}_{\text{mix}}) \tag{8}$$

which is based on the auto-regressive form, the joint policy mixture, and unbalanced cross-entropy to drive exploration.

4.2 Filtering-based Exploration Constraint

Maximizing $\widehat{\mathcal{J}}^{\text{ar}}_{\text{uce}}(\boldsymbol{\pi}_\theta)$ increases the joint policy diversity and thus helps agents to find a new joint policy that is different from previous joint policies. However, not all joint policies in the joint policy space are helpful for exploration. For example, under the guidance of diversity, agents will even learn a new joint policy that always stays in a certain state. This case may happen as long as the new joint policy differs from the previous ones. Obviously, such joint policies would waste opportunities that could be used to explore new states. To this end, it is necessary to add constraints when maximizing $\widehat{\mathcal{J}}^{\text{ar}}_{\text{uce}}(\boldsymbol{\pi}_\theta)$. One natural way is to consider in Eq. 8 the effect of each observation-action $(\boldsymbol{o}_t, \boldsymbol{a}_t)$ pair on the exploration, i.e, the uncertainty (or novelty) of s_{t+1}. Therefore, we modify Eq. 8 as follows

$$\mathbb{E}_{\tau\sim\pi_\theta}\big[\sum_t d(\boldsymbol{o}_t, \boldsymbol{a}_t) u(s_{t+1})\big] \tag{9}$$

where $d(\boldsymbol{o}_t, \boldsymbol{a}_t) = \log \frac{1}{\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)} + \mathbb{I}\Big[\frac{\boldsymbol{\pi}_{\text{rand}}(\boldsymbol{a}_t|\boldsymbol{o}_t)}{\boldsymbol{\pi}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)} \leq 1\Big] \log\beta$, and $u(s_{t+1})$ is the uncertainty of s_{t+1}.

However, the constraint is too mild, and agents can still get positive incentives from Eq. 9 by visiting states with low uncertainty $u(s_{t+1})$. To this end, we introduce a *filtering-based exploration constraint*

$$\widehat{\mathcal{J}}^{\text{ar}}_{\text{fec,uce}}(\boldsymbol{\pi}_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\big[\sum_t d(\boldsymbol{o}_t, \boldsymbol{a}_t)\mathbb{I}[u(s_{t+1}) \geq c_u] u(s_{t+1})\big] \tag{10}$$

where c_u is a hyperparameter that controls the strength of the constraint. Compared to Eq. 9, the above constraint is stronger, and agents can only get positive incentives when they visit states with uncertainty not less than c_u. In general, by combining the joint policy diversity with the exploration constraint, we encourage agents to find a new joint policy that is helpful for exploration.

We maximize $\widehat{\mathcal{J}}^{\text{ar}}_{\text{fec,uce}}(\boldsymbol{\pi}_\theta)$ by using the bonus $b_{\text{jpd}}(\boldsymbol{o}_t, \boldsymbol{a}_t)$ which is defined as

$$\begin{aligned} b_{\text{jpd}} = \mathbb{I}[u(s_{t+1}) \geq & c_u] u(s_{t+1}) \Bigg(\log \frac{1}{\pi^{\text{ar}}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)} \\ & + \mathbb{I}\left[\left(\frac{\pi^{\text{ar}}_{\text{rand}}(\boldsymbol{a}_t|\boldsymbol{o}_t)}{\pi^{\text{ar}}_{\text{mix}}(\boldsymbol{a}_t|\boldsymbol{o}_t)}\right) \leq 1\right] \log\beta \Bigg) \end{aligned} \tag{11}$$

4.3 Overall Exploration Bonus

In this section, we introduce the final augmented reward $\hat{r}$, which includes the extrinsic reward r, the bonus b_{jpd} based on the constrained joint policy diversity, and the bonus b_{cls} which

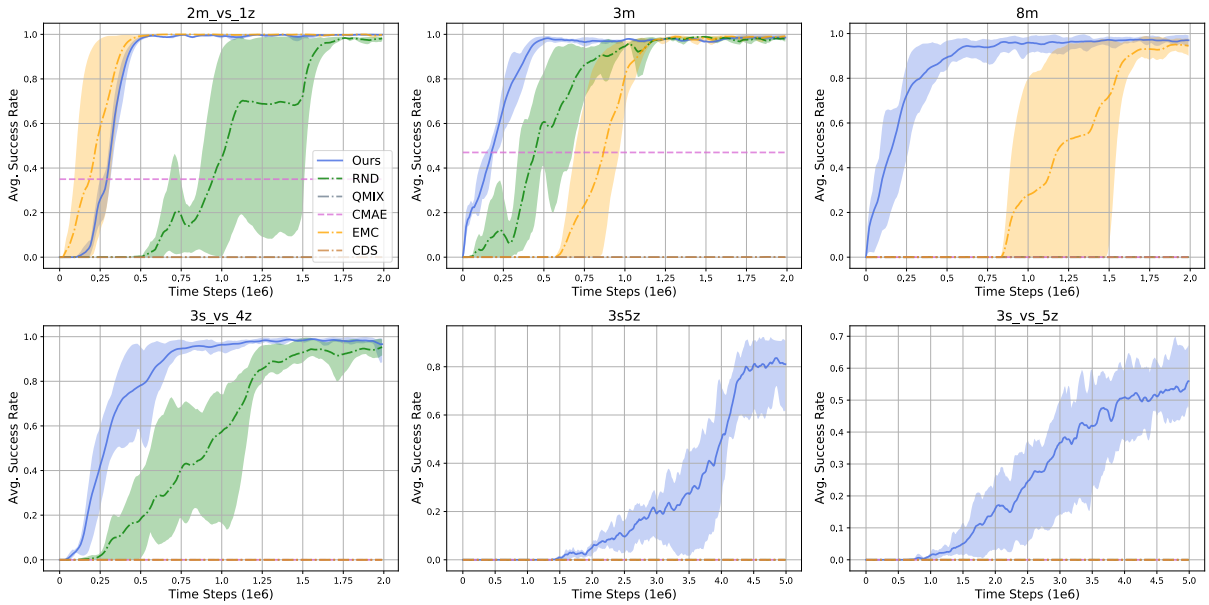

Figure 4: Comparison of our method against baselines on the sparse-reward version of SMAC. The proposed method significantly outperforms all baseline methods.

is the classical exploration bonus based on state uncertainty. In this paper, we use easy-to-implement count-based bonus as our $b_{\text{cls}} = 1/\sqrt{N_{\text{cntr}}(s_{t+1})}$, where $N_{\text{cntr}}(\cdot)$ stands for the state visit count. For simplicity, we replace $u(s_{t+1})$ in Eq. 11 with b_{cls}. For continuous state tasks, such as SMAC, we use RND [Burda *et al.*, 2018b] to estimate b_{cls}. We use the addition operation that is widely used in the literature [Zha *et al.*, 2021; Wang *et al.*, 2019] to combine b_{jpd} and b_{cls}. Formally, the final augmented reward received by agents at each timestep is defined as

$$\hat{r} = r + w_1 \cdot b_{\text{jpd}} + w_2 \cdot b_{\text{cls}} \tag{12}$$

where w_1 and w_2 are hyperparameters.

5 Experiments

The experiments are designed to answer the following research questions: **RQ1**: how is our method compared with the state-of-the-art exploration methods on multi-agent benchmarks in the sparse-reward setting (Sec. 5.1)? **RQ2**: how will our method explore environments without extrinsic rewards (Sec. 5.2)? **RQ3**: how important each component is in our method (Sec. 5.3)? We evaluate our method 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. All experiments run with five random seeds. Details for environments and training are given in supplementary material.

Experimental Setup. In MPE, following previous works [Wang *et al.*, 2019; Liu *et al.*, 2021], we consider three standard tasks: `Push_Box`, `Pass`, `Secret_Room`. We also consider other challenging tasks: `Room`, `Large_Push_Box` and `Large_Pass`. In all tasks, agents see a positive reward only when they complete the given task. In SMAC, we consider six standard tasks: `2m_vs_1z`, `3m`, `8m`, `3s_vs_4z`, `3s5z`, and `3s_vs_5z`. We consider the sparse reward setting, which means agents see a positive reward only when all enemies are taken care of. In GRF [Kurach *et al.*, 2020], following previous 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. Following previous works [Liu *et al.*, 2021; Wang *et al.*, 2019], we first consider **QMIX** [Rashid *et al.*, 2018] which is a popular value-based method for MARL, and **COUNT** which includes a classical count-based bonus [Strehl and Littman, 2008] on top of QMIX. Then, we consider recently proposed methods: **CMAE** [Liu *et al.*, 2021] utilizes domain knowledge and 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 exploration; **CDS** [Chenghao *et al.*, 2021] introduces diversity between agents to encourage extensive exploration. In SMAC and GRF, the count-based bonus and our b_{cls} are approximated by RND [Burda *et al.*, 2018b]. Our method is based on COUNT and introduces additional constrained joint policy diversity.

5.1 Experiments on Standard Multi-agent Tasks

Results on MPE. We first compare our method with baselines on MPE tasks. The training curves are included in

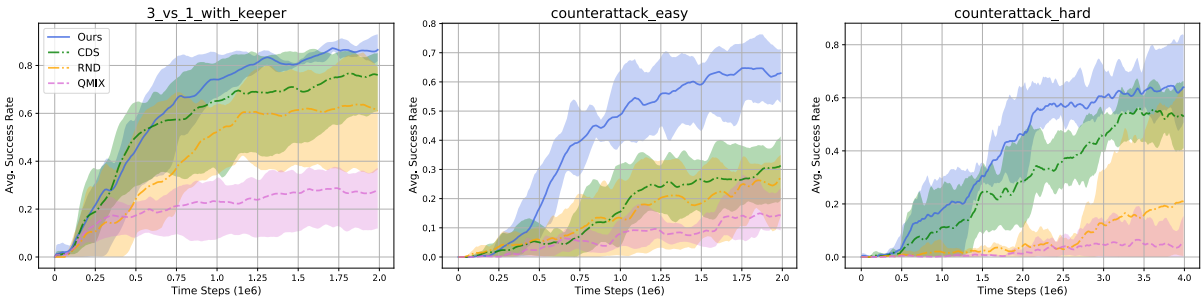

Figure 5: Comparison of our method against baselines on the sparse-reward version of Google Research Football.

Task (1M)	COUNT	CMAE	Ours w/o JPD	Ours
Push_Box	100.2K±13.4K	108.9K±19.7K	129.4K±28.9K	**150.6K±8.2K** (+50%)
Pass	164.8K±7.7K	133.9K±22.2K	204.1K±31.9K	**259.7K±20.6K** (+58%)
Secret_Room	99.3K±9.6K	78.5K±7.1K	126.4K±5.9K	**145.8K±9.2K** (+47%)
Room	483.2K±22.4K	236.7K±14.1K	498.8K±50.7K	**574.4K±16.2K** (+19%)
Large_Push_Box	146.6K±11.2K	160.1K±19.5K	177.6K±25.8K	**234.7K±26.3K** (+60%)
Large_Pass	467.7K±21.6K	182.3K±46.4K	510.1K±9.9K	**524.8K±9.9K** (+12%)

Table 1: The number of explored states for each methods trained without extrinsic reward. Higher number is better for exploration.

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. CMAE which utilizes domain knowledge is the state-of-the-art method in sparse-reward MPE. We observe that CMAE can learn winning strategies on simple tasks but the performance is unstable, while on difficult tasks CMAE fails. As we expected, COUNT which combines with a count-based exploration bonus can solve easy tasks, but does not perform well on hard tasks. Our method which introduces constrained joint policy diversity on top of COUNT shows amazing sample efficiency on all tasks. Compared to COUNT, our method solves tasks faster on all tasks, which confirms the importance of constrained joint policy diversity. Compared to CMAE, our method achieves comparable performance on some tasks (e.g., `Secret_Room`), and our method still achieve good performance on other tasks (e.g., `Large_Pass`) where CMAE fails. This demonstrates the stability of our method.

Results on SMAC. To further study **RQ1**, we evaluate our method in more challenging tasks with continuous state space. We consider the sparse reward setting, which means agents see a positive reward only when all enemies are taken care of. Since CMAE does not provide an implementation on SMAC, we get the results of CMAE from the original paper [Liu *et al.*, 2021]. As shown in Fig. 4, QMIX which relies on random exploration fails in all tasks. And CDS, which introduces diversity between agents, also can not learn a winning strategy. RND and EMC can solve easy tasks, such as `2m_vs_1z` and `3m`. However, as we expected, they both fail in the more challenging tasks such as `3s_vs_5z` which is classified as hard even in the dense-reward setting [Samvelyan *et al.*, 2019]. In contrast, our method works well in all tasks. Specifically, on easy tasks, our method significantly outperforms RND and achieves comparable performance to EMC. In hard `3s_vs_5z`, to our knowledge, our method is the first to learn winning strategies without domain knowledge under the sparse-reward setting.

Results on GRF. Next, we evaluate our method 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 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 method become more obvious. On easy tasks, our method slightly outperforms the state-of-the-art method CDS, but on harder tasks, our method significantly outperforms all baselines.

5.2 Exploration without Extrinsic Rewards

To study **RQ2** and analyze exploration behaviors of the proposed method, we train COUNT, CMAE, Ours w/o JPD, and Ours after 1 million without extrinsic rewards. Ours w/o JPD ablates the joint policy diversity by only keeping the filtering-based exploration constraint $\mathbb{I}[u(s_{t+1}) \geq c_u]u(s_{t+1})$ in Eq. 11. We first report the number of explored states and the improvement of our method compared to COUNT in Tab. 1. More explored states mean better exploration. We also report the visited state entropy in the supplementary material. We make four observations. First, our method explores more states than COUNT in all tasks, showing that the superior performance (Fig 3) in the sparse-reward setting comes from our better exploration. Second, our method outperforms Ours w/o JPD, which confirms the importance of joint policy diversity. Third, the filtering-based exploration constraint is beneficial for exploration. This may be because assigning additional bonuses to states with higher uncertainty (i.e., higher novelty) allows agents parameterized by neural networks to more

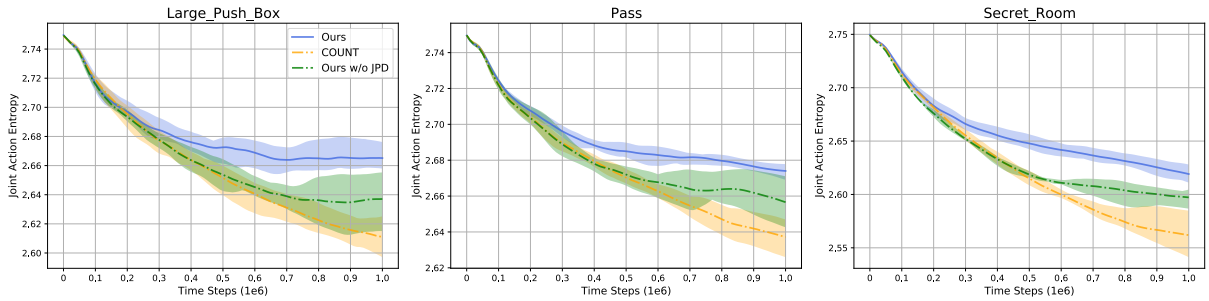

Figure 6: All agents are trained without extrinsic rewards. The y-axis is the joint action entropy of the joint policy mixture. A low joint action entropy is harmful to exploration.

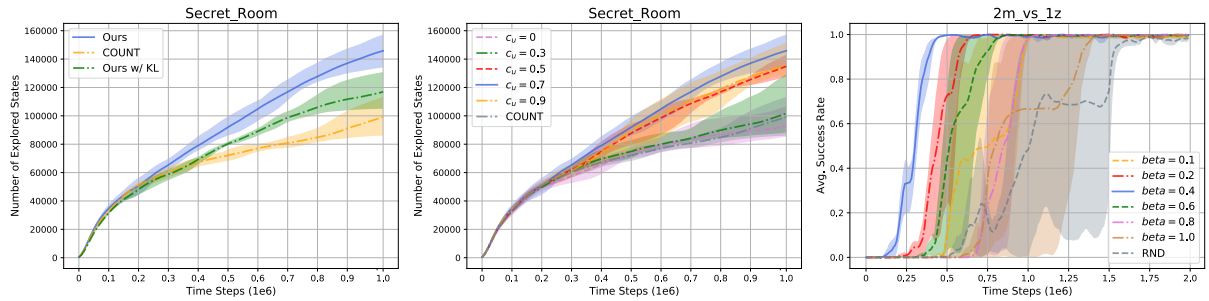

Figure 7: Ablation studies for the proposed method.

easily distinguish between novel states and familiar states. Fourth, states explored by CMAE are significantly less than that of our method and COUNT. This suggests that the strong performance of CMAE in some tasks comes from its use of domain knowledge, rather than its exploration ability.

In supplementary material, we report the percentage of states with higher novelty (uncertainty greater than 0.7) in an episode. We observe that COUNT wastes many opportunities that could be used to explore new states. In contrast, our method has better exploration ability and wastes fewer opportunities.

To further understand the impact of the joint policy diversity on exploration, the joint action entropy of the joint policy mixture, which is defined in Eq. 4, is shown in Fig. 6. We first observe that the joint action entropy of our method is higher than that of Ours w/o JPD. A high joint action entropy implies that the action selection of the joint policy mixture, which consists of historical joint policies, is more uniform. This explains why our method can explore more states than Ours w/o JPD, and confirms the importance of the diversity term in Eq. 11. In addition, we also observe that the exploration constraint can slightly increase the joint action entropy. This may be a by-product of the rapid change in bonuses caused by the exploration constraint.

5.3 Ablations

To study **RQ3** and further understand the proposed method, we carry out ablation studies. To confirm that our unbalanced cross-entropy is more suitable for exploration than KL-divergence, we replace the unbalanced cross-entropy in Eq. 11 with KL-divergence. As shown in Fig. 7 (left), our method is better under the reward-free setting.

Then, to study the impact of c_u (Eq. 10) on exploration, we train agents with different c_u under the reward-free setting. The results are reported in Fig. 7 (middle). We make two observations. First, a weak constraint ($c_u = 0$) does not improve exploration compared to COUNT. This confirms the importance of the proposed filtering-based constraint. Second, a too-strong constraint ($c_u = 0.9$) is also harmful. This is because it would make $b_{\text{jpd}} = 0$ (Eq. 11) in most cases.

To study the impact of β in Eq. 6, we train agents with different β in `2m_vs_1z` under the sparse-reward setting. The results are shown in Fig. 7 (right). We make two observations. First, the unbalanced cross-entropy is significantly better than the original cross-entropy ($\beta = 1.0$). This confirms that the unbalanced cross-entropy is important for efficient exploration. Second, a too-low β (such as 0.1) also hurts sample efficiency. This is reasonable because in this case, the penalty term overwhelms the cross-entropy term.

6 Conclusion

In this paper, we focus on sparse-reward multi-agent tasks. We present a novel exploration method, which exploits both constrained joint policy diversity and classical bonus based on state uncertainty to jointly encourage exploration. We evaluate our method on three challenging environments under the sparse-reward setting. Results show that our method pushes forward state-of-the-art. One limitation of the current work is that it focuses on discrete-action tasks. In the future, we will explore the possibility of extending our method to scenarios with continuous-action space.

Acknowledgements

This work is supported in part by Basic Cultivation Fund project, CAS (JCPYJJ-22017) and the Youth Innovation Promotion Association CAS.

References

[Azar *et al.*, 2017] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

[Badia *et al.*, 2020] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

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

[Bellemare *et al.*, 2016] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[Burda *et al.*, 2018a] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[Burda *et al.*, 2018b] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[Chenghao *et al.*, 2021] Li Chenghao, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[Fu *et al.*, 2022] Wei Fu, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. Revisiting some common practices in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2206.07505*, 2022.

[Hazan *et al.*, 2019] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.

[Jin *et al.*, 2018] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

[Kurach *et al.*, 2020] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4501–4510, 2020.

[Liu *et al.*, 2021] Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 6826–6836. PMLR, 2021.

[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*, 2017.

[Mahajan *et al.*, 2019] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *arXiv preprint arXiv:1910.07483*, 2019.

[Pathak *et al.*, 2017] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

[Rashid *et al.*, 2019] Tabish Rashid, Bei Peng, Wendelin Boehmer, and Shimon Whiteson. Optimistic exploration even with a pessimistic initialisation. In *International Conference on Learning Representations*, 2019.

[Ryu *et al.*, 2022] Heechang Ryu, Hayong Shin, and Jinkyoo Park. Remax: Relational representation for multi-agent exploration. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1137–1145, 2022.

[Samvelyan *et al.*, 2019] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[Strehl and Littman, 2008] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[Swamy *et al.*, 2020] Gokul Swamy, Siddharth Reddy, Sergey Levine, and Anca D Dragan. Scaled autonomy: Enabling human operators to control robot fleets. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5942–5948. IEEE, 2020.

[Wang *et al.*, 2019] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent explo-

ration. In *International Conference on Learning Representation*, 2019.

[Xu *et al.*, 2021] Pei Xu, Qiyue Yin, Junge Zhang, and Kaiqi Huang. Deep reinforcement learning with part-aware exploration bonus in video games. *IEEE Transactions on Games*, 14(4):644–653, 2021.

[Zha *et al.*, 2021] Daochen Zha, Wenye Ma, Lei Yuan, Xia Hu, and Ji Liu. Rank the episodes: A simple approach for exploration in procedurally-generated environments. *arXiv preprint arXiv:2101.08152*, 2021.

[Zhang *et al.*, 2021] Qizhen Zhang, Chris Lu, Animesh Garg, and Jakob Foerster. Centralized model and exploration policy for multi-agent rl. *arXiv preprint arXiv:2107.06434*, 2021.

[Zheng *et al.*, 2021] Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34, 2021.