Self-Supervised Multi-Agent Diversity with Nonparametric Entropy Maximization

Tianxu Li
College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics
Nanjing, China
tianxuli@nuaa.edu.cn

Kun Zhu*
College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics
Nanjing, China
zhukun@nuaa.edu.cn

ABSTRACT

Learning decentralized policies for agents has drawn increasing interest in recent works to solve the scalability issue that arises in Multi-Agent Reinforcement Learning (MARL), where all agents may share the parameters of a policy network to make action decisions. However, such parameter sharing can hinder efficient exploration, as some agents may learn similar behaviors. Unlike previous fully-supervised mutual information-based methods that encourages multi-agent diversity, in this paper, we propose a novel multi-agent exploration method called Contrastive Trajectory Entropy Maximization (CTEM). Our method adopts a non-parametric entropy estimator to maximize the entropy of trajectories of different agents in a self-supervised contrastive representation space, leading to diverse policies and sufficient exploration. Such an entropy estimator avoids complex density modeling and scales well in high-dimensional multi-agent environments. We deploy our method in MARL by introducing an intrinsic reward for agents to achieve entropy maximization. To demonstrate the effectiveness of our method, we conduct experiments on multiple challenging MARL benchmark tasks. Our method yields superior performance than existing state-of-the-art methods.

KEYWORDS

Multi-Agent Reinforcement Learning, Exploration, Multi-agent diversity, Trajectory Entropy Maximization

ACM Reference Format:
Tianxu Li and Kun Zhu. 2025. Self-Supervised Multi-Agent Diversity with Nonparametric Entropy Maximization. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Multi-Agent Reinforcement Learning (MARL) has shown promise in solving various multi-agent tasks such as multiplayer video games [30] and traffic light control [35], which has drawn increasing attention in recent years. MARL enables efficient cooperation by jointly training multiple agents to maximize the team returns. However, learning efficient cooperative policies for agents in challenging multi-agent tasks still remains a challenge due to the partial observation constraint and the high scalability demand. A common-used framework in recent works to resolve these issues is Centralized Training with Decentralized Execution (CTDE) [17], where each agent makes action decisions based on its local observation using a decentralized policy that is jointly trained leveraging global information in order to achieve robust and stable performance.

The CTDE framework learns an individual decentralized policy for each agent. However, training a large number of policy network parameters can be inefficient. The parameter sharing technique has been widely adopted in the CTDE framework, allowing all agents to share the same policy network parameters when making action decisions. Consequently, parameter sharing significantly reduces the amount of policy network parameters, efficiently reducing computational complexity and accelerating training speed. Moreover, parameter sharing enables sharing of experience among agents during centralized training, which not only contributes to the learning of a robust and stable policy but also enhances the overall learning efficiency [32].

Benefiting from these advantages, a variety of MARL algorithms have incorporated the parameter sharing technique, including value-decomposition methods [9, 23, 29, 31, 36] and policy gradients [18, 21, 34, 38]. Unfortunately, the agents sharing the policy network parameters easily learn homogeneous behaviors since the agents tend to behave similarly under similar observations [8], impeding the emergence of multi-agent diversity and efficient exploration. Challenging multi-agent tasks typically require extensive exploration and diversified policies among agents. To illustrate, consider a football game where agents must collaboratively work towards scoring a goal. When agents exhibit uniform policies, they may compete for the ball, consequently resulting in ineffective competition. To win the game, the agents need to learn diverse policies and play different roles to effectively pass the ball.

Several methods [2, 10, 12, 14] have been proposed to encourage identity-aware multi-agent diversity in a fully-supervised manner by maximizing the mutual information between the trajectories and agent identities. These methods aim to distinguish the trajectories of different agents according to the agent identities. However, despite their achievements, this category of methods easily falls into local optimum since the agents prefer to visit known trajectories that contain more identity information and do not explore comprehensively. As a result, the agent trajectories may overfit agent identities.

In this paper, we propose a novel exploration method called Contrastive Trajectory Entropy Maximization (CTEM) to promote

*Corresponding author. The authors are also with Collaborative Innovation Center of Novel Software Technology and Industrialization.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

multi-agent diversity in a self-supervised manner while guaranteeing efficient exploration. Unlike previous work, we adopt neither mutual information nor a trajectory discriminator. The intuition behind our method is that the agents need to explore the environment sufficiently to visit any states where they might get rewards. To achieve this goal, our method relies on maximizing the entropy of trajectories of different agents. Since it is intractable to maximize the entropy in the high-dimensional trajectory space as the density model of the agent's trajectory is unknown, our method instead employs a nonparametric particle-based entropy estimator [1, 26] that is asymptotically unbiased for the entropy. The particle-based entropy estimator calculates the mean Euclidean distance between a particle and its neighbors. To make the distance meaningful, we construct a contrastive representation space by encoding trajectory space to a low-dimensional representation space using self-supervised contrastive learning [3]. Our method can be applied to MARL algorithms by introducing an intrinsic reward for the agent to maximize the entropy. The contributions of this work can be summarized as follows: first, we propose a novel self-supervised exploration method called CTEM to encourage multi-agent diversity by maximizing the trajectory entropy based on a nonparametric particle-based entropy estimator in a contrastive representation space; second, we evaluate our method in various challenging multi-agent tasks. The experimental results demonstrate the significant outperformance of our method compared to other existing state-of-the-art MARL algorithms.

2 BACKGROUNDS

We consider modeling the fully cooperative multi-agent tasks as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [22] defined as a tuple $\langle A, S, U, P, R, O, \Omega, \gamma \rangle$, where $A = \{1, \ldots, |A|\}$ denotes a set of $|A|$ agents, $s \in S$ represents the environment state, and U is the set of actions. At each time step, each agent a receives an observation $o^a \in \Omega$ drawn from the function $O(s, a)$ and subsequently selects an action $u^a \in U$. The selected actions of all agents form a joint action denoted as $\boldsymbol{u}$. The environment then transitions to a new state s' with the probability defined by the transition function $P(s' \mid s, \boldsymbol{u})$. Simultaneously, the environment provides a shared reward $r = R(s, \boldsymbol{u})$ for the agents. $\gamma \in [0, 1)$ serves as a reward discount factor. The trajectory of each agent is denoted as $\tau^a \in \mathcal{T}$ that is composed of observation-action history. Each agent learns a decentralized policy $\pi^a(u^a \mid \tau^a)$, assembling a joint policy $\boldsymbol{\pi}$, towards maximizing a joint action-value function $Q^{\boldsymbol{\pi}}(s, \boldsymbol{u}) = \mathbb{E}_{s_{0:\infty}, \boldsymbol{u}_{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, \boldsymbol{u}_0 = \boldsymbol{u}, \boldsymbol{\pi}\right]$.

3 THE LIMITATION OF MUTUAL INFORMATION-BASED METHODS

One of the common approaches to encourage multi-agent diversity is to maximize the mutual information between trajectories and agent identities [2, 10, 12, 14]. However, these works share a limitation that the agents are likely to prefer known trajectories that contain more identity information than novel trajectories, resulting in inefficient exploration. We next analyze this limitation from a theoretical standpoint. We present the reward functions associated with the exploration of both familiar and new trajectories. The theoretical results reveal that agents attain higher rewards when visiting known trajectories compared to their exploration of new trajectories.

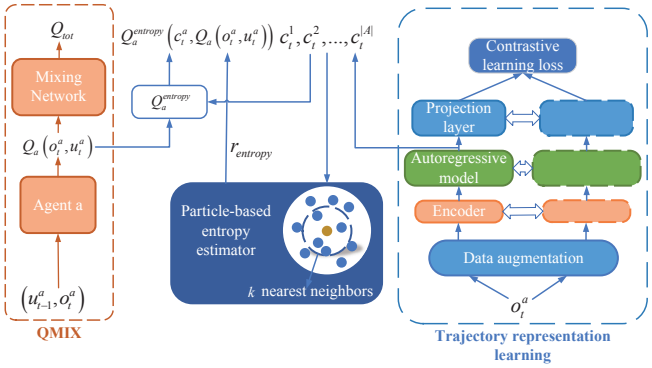


Figure 1: Diagram of our proposed CTEM.

Given the mutual information between the trajectory τ and agent identity i as

$$\begin{aligned} I(i; \tau) &= \mathbb{E}_{i,\tau}[\log p(i \mid \tau)] - \mathbb{E}_i[\log p(i)] \\ &\geq \mathbb{E}_{i,\tau}\left[\log q_\theta(i \mid \tau)\right] - \mathbb{E}_i[\log p(i)] \end{aligned} \tag{1}$$

where the unknown posterior distribution $p(i \mid \tau)$ is approximated by a variational distribution $q_\theta(i \mid \tau)$. We parameterize $q_\theta(i \mid \tau)$ with θ and update θ towards maximizing the likelihood on (i, τ)-tuples stored in the replay buffer. Prior works maximize the mutual information by employing the variational lower bound as an intrinsic reward

$$\begin{aligned} r(\tau, i') &= \log q_\theta(i' \mid \tau) - \log p(i') \\ &= \log q_\theta(i' \mid \tau) + \log |A| \end{aligned} \tag{2}$$

where $i' \sim p(i)$, a uniform distribution, thus $-\log p(i') = \log |A|$, where $|A|$ is the number of agents. We assume access to a perfect distribution $q_\theta(i \mid \tau)$, yielding $\sum_{a=1}^{|A|} q_\theta(i_a \mid \tau) = 1$.

Intrinsic reward for known trajectories The intrinsic reward function motivates the agents to visit known trajectories τ where $q_\theta(i' \mid \tau) \to 1$. As a result,

$$r_{\max} = \log 1 + \log |A| = \log |A|. \tag{3}$$

Intrinsic reward for new trajectories For new trajectories, $q_\theta(i' \mid \tau)$ is unknown. Here, we assign a null probability to unseen trajectories by adding a background class to the model. The penalization received by agents when they visit unseen trajectories is

$$r'_{\text{new}} = \lim_{q_\theta(i'|\tau) \to 0} \log q_\theta(i' \mid \tau) + \log |A| = -\infty \tag{4}$$

We note that when the distribution $q_\theta(i' \mid \tau)$ converges, the agents can achieve larger rewards for visiting known trajectories than exploring new trajectories.

4 CONTRASTIVE TRAJECTORY ENTROPY MAXIMIZATION

To resolve this limitation, in this paper, our aim is to encourage multi-agent diversity by maximizing the entropy of trajectories of different agents in an abstract representation space, unlike the previous mutual information maximization method. First, we map the trajectory space to a latent contrastive representation space with a self-supervised contrastive learning method. Then, we propose a novel nonparametric method to maximize the trajectory entropy by introducing per-agent intrinsic rewards.

4.1 Learning Contrastive Trajectory Representation

We propose a novel trajectory representation learning method based on contrastive learning. Recent work [13, 28] shows promise in using contrastive learning to learn meaningful representations in RL. The reason we choose contrastive learning is that we aim to distinguish trajectories of different agents based on the distance between them in the representation space to ensure that our entropy maximization objective works properly.

Concretely, our representation learning method is based on the contrastive learning loss employed in simCLR [3]. Different from simCLR that is designed to learn representations of images, we propose a novel structure to learn distinguishable representations of different agents' trajectory sequences. First, we randomly sample a batch of trajectories of all agents from the replay buffer $\{\tau^a\}_{a=1}^{|A|}$. Then, we apply the data augmentation operation, denoted as $aug(\cdot)$, that adds Gaussian noise randomly sampled from the normal distribution $\mathcal{N}\left(0, 0.1^2\right)$ to each observation o_t^a in the trajectory sequence τ^a, improving the robustness and generalization of the representation learning model. We augment each observation o_t^a two times to obtain a key $(o_t^a)_k = aug(o_t^a)$ and a query $(o_t^a)_v = aug(o_t^a)$, which are then encoded into a latent representation space by an encoder $z_t^a = g_{\theta_e}(\cdot)$, respectively. Next we summarize the latent representations with an autoregressive model g_{θ_g} to a trajectory representation $c_t^a = g_{\theta_g}(z_{\leq t}^a)$, which alleviates the non-stationary issue caused by partial observability constraints in multi-agent environments and leads to more stable policies. For simplicity, we denote $g_\theta = \left\{g_{\theta_e}, g_{\theta_g}\right\}$. Note that g_θ only encodes the observations of agents since we hope to motivate agents to explore diverse observations via entropy maximization. The trajectory representation is then input to a projection network h_ϕ to obtain a final output where the contrastive learning loss is imposed. We train the network g_θ and the projection network h_ϕ by minimizing the contrastive learning loss:

$$\ell(\theta,\phi) = -\frac{1}{2|A|}\sum_{a=1}^{|A|}\left[\log\frac{\exp\left(\text{sim}\left(h_\phi\left(g_\theta\left((o_t^a)_k\right)\right), h_\phi\left(g_\theta\left((o_t^a)_v\right)\right)\right)\right)}{x+y}\right],$$

$$\text{where}\quad x = \sum_{a'=1}^{|A|}\mathbf{1}_{[a\neq a']}\exp\left(\text{sim}\left(h_\phi\left(g_\theta\left((o_t^a)_k\right)\right), h_\phi\left(g_\theta\left((o_t^{a'})_k\right)\right)\right)\right),$$

$$y = \sum_{a'=1}^{|A|}\mathbf{1}_{[a\neq a']}\exp\left(\text{sim}\left(h_\phi\left(g_\theta\left((o_t^a)_k\right)\right), h_\phi\left(g_\theta\left((o_t^{a'})_v\right)\right)\right)\right). \tag{5}$$

In Equation 5, $\mathbf{1}_{[a\neq a']}$ is an indicator function evaluating to 1 iff $a \neq a'$ and $\text{sim}(u, v)$ is the cosine similarity between u and v. The goal of contrastive learning loss shown in Equation 5 is to guarateen that the key $(o_t^a)_k$ is more close to the query $(o_t^a)_v$ than other key-query points $\left\{(o_t^{a'})_k, (o_t^{a'})_v\right\}_{a'=1,a'\neq a}^{|A|}$ in the contrastive representation space.

In practice, for simplicity, we employ resnet blocks [7] for the encoder and a GRU unit [4] for the autoregressive model. It is notable that the trajectory representation learning relies on a complete self-supervised method to achieve distinguishability among agents without using explicit pre-defined identities of agents, e.g., labeling agents with one-hot vectors like prior works [2, 10, 12, 14]. We find it helps to use the self-supervised method to improve the exploration of the MARL algorithm.

4.2 Nonparametric Entropy Maximization

Maximizing entropy using density estimation like the previous work [6] is non-trivial and challenging in high-dimensional multi-agent settings. To maximize the entropy of trajectory representations of different agents, our method uses a nonparametric particle-based entropy estimator [1, 26] that has been widely investigated in statistics [11]. The particle-based entropy estimator gives the measurement of the sparsity of data distribution depending on the distance between the sampled data point and its k-th nearest neighbor point.

We then present the implementation of the particle-based entropy estimator in our method. In this paper, we treat each trajectory representation as a particle. Concretely, given a set of trajectory representations of all agents $\left\{c_t^a \in \mathbb{R}^d\right\}_{a=1}^{|A|}$ learned by g_θ, the particle-based entropy estimator can be defined as

$$\mathcal{H}(c_t) = -\frac{1}{|A|}\sum_{a=1}^{|A|}\log\frac{k}{|A|\mathrm{v}_a^k} + b(k) \propto \sum_{a=1}^{|A|}\log \mathrm{v}_a^k, \tag{6}$$

where $b(k)$ works as a bias correction depending on the hyperparameter k, and v_a^k is the volume of a hypersphere with a radius of $\left\|c_t^a - (c_t^a)^{(k)}\right\|$,

$$\mathrm{v}_a^k = \frac{\left\|c_t^a - (c_t^a)^{(k)}\right\|^d \cdot \pi^{d/2}}{\Gamma\left(d/2+1\right)} \tag{7}$$

where $(c_t^a)^{(k)}$ is the k-th nearest neighbor of c_t^a in set $\{c_t^a\}_{a=1}^{|A|}$, the operator $\|\cdot\|$ is used to calculate the Euclidean distance, and Γ is the gamma function. Intuitively, v_a^k serves as an indicator of the sparsity around the trajectory representation of each agent and the entropy estimator $\mathcal{H}(c_t)$ quantifies the average volume around the trajectory representation of each agent.

Given the definition of v_a^k, we can thus simplify the particle-based entropy estimator in Equation 6 as follows:

$$\mathcal{H}(c_t) \propto \sum_{a=1}^{|A|}\log\left\|c_t^a - (c_t^a)^{(k)}\right\|^d \tag{8}$$

where $\mathcal{H}(c_t)$ is proportional to the sum of the log of the Euclidean distance between the trajectory representation and its k-th nearest

neighbor. However, we empirically observe that using the entropy estimator given by Equation 8 easily leads to learning unstable policies. To ensure proper operation in multi-agent settings, we present a novel entropy estimator that calculates the average value of all k nearest neighbors around the trajectory representation:

$$\mathcal{H}(c_t) := \sum_{a=1}^{|A|} \log\left(b + \frac{1}{k} \sum_{(c_t^a)^{(j)} \in \mathrm{N}_k\left(c_t^a\right)} \left\| c_t^a - (c_t^a)^{(j)} \right\|^d\right), \quad (9)$$

where $\mathrm{N}_k\left(c_t^a\right)$ is a set of k nearest neighbors around a trajectory representation c_t^a. We additionally introduce a constant b for numerical stability that is set to 1 for all experiments.

In order to encourage multi-agent diversity by maximizing the entropy $\mathcal{H}(c_t)$, we can treat the entropy as an intrinsic reward $r_{entropy}^a$, where the representation of o_{t+1}^a is treated as a particle in the entropy. Concretely, given a transition $(o_t^a, u_t^a, o_{t+1}^a)$ of agent a, we define the intrinsic reward function for agent a as follows:

$$r_{entropy}^a = \log\left(b + \frac{1}{k} \sum_{g_\theta(o_{t+1}^a)^{(j)} \in \mathrm{N}_k\left(g_\theta(o_{t+1}^a)\right)} \left\| g_\theta(o_{t+1}^a) - g_\theta(o_{t+1}^a)^{(j)} \right\|^{|A|}\right). \quad (10)$$

Intuitively, the intrinsic reward incentivizes agents to visit diverse trajectories that have larger distances in contrastive representation space. For the pytorch-style pseudocode of CTEM, we refer the reader to Technical Appendix 3 in the supplemental material. We also provide the source code of our method in the supplemental material.

Differences to previous methods Note that our objective is completely different from prior methods [2, 10, 12, 14] that maximize the objective of mutual information between trajectories τ and agent identities i by introducing an intrinsic reward

$$r_{MI}(\tau, i) = \log q_\theta(i \mid \tau) - \log p(i) \quad (11)$$

where $q_\theta(i \mid \tau)$ is a variational distribution that is trained towards maximizing the likelihood on (i, τ)-tuples stored in the replay buffer, and $p(i)$ is a fixed uniform distribution. The above intrinsic reward r_{MI} encourages the agents to visit trajectories that contain more identity information. In contrast, our intrinsic reward $r_{entropy}^a$ given by Equation 10 encourages agents to visit diverse trajectories with larger distances in contrastive representation space, leading to entropy maximization.

4.3 Learning Algorithm

In this section, we introduce how to integrate our algorithm with QMIX [23], a value-decomposition-based MARL algorithm. In QMIX, each agent learns its individual policy through optimizing an approximation Q_{tot} for the joint action-value function $Q^{\boldsymbol{\pi}}$. QMIX monotonically mixes the agent utilities (where the agents' policies are derived) of all agents with a mixing network to output the Q_{tot}. To implement our method on top of QMIX, we introduce an additional intrinsic reward for each agent and simultaneously learn an intrinsic utility network $Q_a^{entropy}$ for each agent a towards maximizing the total intrinsic rewards, yielding entropy maximization. Concretely, the intrinsic utility network $Q_a^{entropy}$ takes as input the agent utility $Q_a(o_t^a, u_t^a)$ and current trajectory representation c_t^a. To learn the intrinsic utility network $Q_a^{entropy}$, we minimize the TD loss with our intrinsic rewards:

$$\begin{aligned} \mathcal{L}_{TD}^{entropy} &= \mathbb{E}_{(o_t^a, u_t^a, o_{t+1}^a)\sim\mathcal{D}}\left[\left(Q_a^{entropy}\left(c_t^a, Q_a(o_t^a, u_t^a)\right) - y\right)^2\right], \\ where \quad y &= r_{entropy}^a + \gamma \bar{Q}_a^{entropy}\left(c_{t+1}^a, \bar{Q}_a\left(o_{t+1}^a, u_{t+1}^a\right)\right). \end{aligned} \quad (12)$$

$\bar{Q}_a^{entropy}$ and $\bar{Q}_a$ are target networks to stablize training. Each time we randomly take a minibatch of trajectory samples from the replay buffer $\mathcal{D}$ for training. Notably, the loss function $\mathcal{L}_{TD}^{entropy}$ introduces an auxiliary gradient to train the agent utility network. The agent utility of QMIX has no actual meaning and constraints, enabling our method to be safely integrated with QMIX. We can thus get the total loss function to learn optimal policies for agents:

$$\mathcal{L}_{total} = \mathcal{L}_{TD}^{QMIX} + \beta \mathcal{L}_{TD}^{entropy}, \quad (13)$$

where $\mathcal{L}_{TD}^{QMIX}$ is the TD loss function of QMIX to learn Q_{tot} and update parameters of agent utility networks towards maximizing team returns. β is a cofficient to change the weight of $\mathcal{L}_{TD}^{entropy}$ compared with $\mathcal{L}_{TD}^{QMIX}$. The overall framework of our method is trained end-to-end in a centralized manner by minimizing $\mathcal{L}_{total}$, where the agent learns its policy towards maximizing both the team returns and the entropy of trajectory representations of different agents. Our method thus promotes multi-agent diversity. Note that our method can also be integrated with policy gradient methods. We refer the reader to Technical Appendix 2 in the supplemental material for the impelementation of our method on top of policy gradient methods.

5 EXPERIMENTS

In this section, we evaluate our proposed CTEM in Pac-Men, SMAC, and SMACv2 benchmarks to demonstrate the superior performance of our proposed method. We compare our proposed CTEM with the state-of-the-art methods, including value-decomposition methods (such as QMIX [23] and QTRAN [27]) and mutual information-based exploration methods (including MAVEN [19], EOI [10], SCDS [14], PMIC [15], LIPO [2], and FoX [12]). For generality, we report both the mean and standard deviation of the performance for CTEM and baselines, derived from five random seeds. For a fair comparison, the hyperparameters across various methods are consistent in each multi-agent task. Hyperparameters and training details are provided in Technical Appendix 5 in the supplemental material.

5.1 Pac-Men

To showcase the effectiveness of our method in promoting multi-agent diversity, we design a grid world environment called Pac-Men, illustrated in Figure 2a, to compare our method to the baselines. In Pac-Men, we initialize four agents positioned in the central room of a maze. Each agent moves in the maze with only partial observability. There are some randomly initialized dots distributed in each edge room. The agents can move to the four edge rooms along paths to eat dots. To intensify the challenge, we set distinct path lengths for each path to the edge rooms. Note that only the

downward path is within the agent's observation scope, which imposes a high demand on efficient exploration.

As shown in Figure 2b, our method achieves significant improvement over QMIX and dramatically outperforms other baselines. QMIX does not learn optimal policies in Pac-Men. To achieve more rewards, the agents require to move to the four edge rooms, respectively, to collect dots. However, the visitation heatmap of QMIX illustrated in Figure 2c demonstrates that some agents learn similar behaviors and go to the same bottom room. These agents may compete for the dots in the same room, resulting in inefficient cooperation. With the help of our method, as shown in Figure 2d, the agents efficiently learn diverse policies and move to the four edge rooms, respectively. This indicates that the objective of entropy maximization promotes the learning of diverse policies. Moreover, we note that the baselines maximizing the mutual information between trajectories and agent identities such as EOI and SCDS achieve similar performance and do not yield satisfactory performance. We argue that this is because these algorithms suffer from insufficient exploration and the agents may not discover the upward room with the longest path. We further present the mutual information-based and our entropy maximization-based intrinsic rewards in Figure 2e, respectively. The results demonstrate that the mutual information-based intrinsic reward does not provide efficient incentives, while our entropy maximization-based intrinsic rewards continuously encourage the agent to explore the optimal cooperative policies.

5.2 SMAC

After initially evaluating our method in a straightforward grid world environment, we advance to a more complex multi-agent setting known as the StarCraft Multi-Agent Challenge (SMAC) [24]. To indicate the effectiveness of our method, we conduct experiments in 6 scenarios of SMAC: 3s5z (easy), 2c_vs_64zg (hard), 7sz (hard), 6h_vs_8z (super hard), corridor (super hard), and 3s5z_vs_3s6z (super hard). Note that performance comparison are not applicable across different SMAC versions. We use the version SC2.4.10 of SMAC to conduct our experiments.

The performance comparison between our proposed CTEM and baselines in the SMAC scenarios is shown in Figure 3. In the three super hard scenarios (6h_vs_8z, corridor, and 3s5z_vs_3s6z), where the strength of enemies is more powerful than that of agents, our method significantly outperforms baselines, indicating that our method is more robust in exploring cooperative policies than baselines by maximizing trajectory entropy. Although QMIX achieves satisfactory performance in the 3s5z and 2c_vs_64zg scenarios, it fails to learn effective policies in other challenging scenarios requiring complex and diverse cooperative policies and needs our method to get better performance.

MAVEN is less efficient in exploring cooperative policies, demonstrating that the trajectory entropy maximization objective yields more efficient exploration than encouraging the visitations of diverse joint behaviors employed in MAVEN. EOI and SCDS achieve promising results in the 3s5z and 2c_vs_64zg scenarios, however, they do not achieve robust results in other more challenging scenarios. We attribute this to the strong mutual dependence between trajectories and agent identities, which impedes further exploration of complex cooperative policies. Similarly, the mutual information-based formation diversity from FoX also suffers from this problem.

We provide the visualization examples of diverse policies learned by our method shown in Figure 6, that emerge in the super hard scenarios (6h_vs_8z, corridor, and 3s5z_vs_3s6z). For example, in the 6h_vs_8z scenario, to cover other agents, one agent quickly moves in the opposite direction to the team. Then most of the enemies are attracted by the agent's movements. The agent keeps kiting the following enemies and draws the most of the enemies' fire. Meanwhile, the few remaining enemies are surrounded by the other agents. As a result, the agents cooperatively distribute the enemies' attacks by taking diverse policies. However, if all the agents take similar behaviors and rush toward enemies, they will soon be defeated by the powerful enemies. We can also note such diverse policies learned by our method in the other two scenarios. These results demonstrate the effectiveness of our method in learning diverse policies, enabling the agents to cooperatively defeat the enemies.

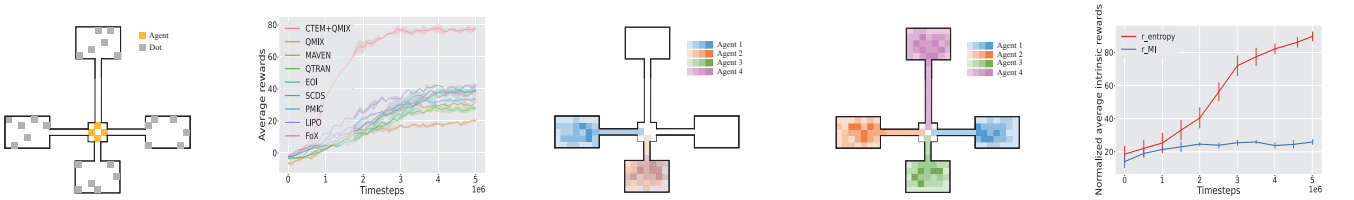
Homogeneous behaviors Notably, our method also achieves superior performance in the easy 3s5z scenario, where agents may sometimes need to behave in the same way in order to master the 'focus fire' trick. We further evaluate our method in similar homogeneous scenarios. The results, presented in Table 1, demonstrates that our method would not impede the learning of homogeneous behaviors that can contribute to more environmental rewards. Our entropy maximization-based method can efficiently balance exploration and exploitation in MARL.

Table 1: Performance of our method and QMIX in homogeneous scenarios.

Method	8m	5m_vs_6m	8m_vs_9m	10m_vs_11m
CTEM+QMIX	0.93±0.03	0.90±0.05	0.91±0.03	0.89± 0.04
QMIX	0.87±0.03	0.65±0.04	0.58±0.05	0.43±0.04

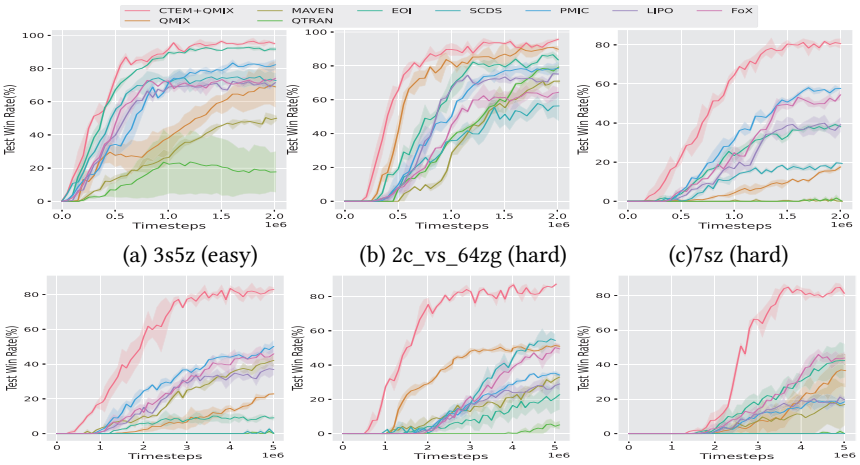
Stochasticity and Exploration Although the scenarios of SMAC are challenging, one limitation of SMAC is that it lacks enough stochasticity in the combat scenarios to test the exploration of MARL algorithms, as the initial positions and compositions of the team are typically fixed. We further adopt a more challenging SMACv2 benchmark [5], enabling stochasticity in the SMAC benchmarks through deploying random start positions and random team compositions in each episode.

We evaluate our method in three scenarios of SMACv2: terran_5_vs_5, protoss_5_vs_5, and zerg_5_vs_5. The experimental results, illustrated in Figure 4, demonstrate that our method performs substantially better than baselines in all the scenarios. Note that QMIX fails to learn optimal policies and lacks sufficient exploration to adapt to the stochasticity in the SMACv2 scenarios. However, by integrating with our method, QMIX significantly improves its performance and learns more exploratory and diverse policies. The mutual information-based baselines such as MAVEN, EOI, and SCDS also fall into local optimum. We argue that this is because the mutual dependence between trajectories and agent identities learned in these algorithms forces agents to visit known trajectories instead of discovering new trajectories. However, our method can continuously explore new trajectories and search for



(a) Illustration of Pac-Men (b) Learning curves (c) Visitation heatmap of QMIX (d) Visitation heatmap of CTEM (e) Normalized average intrinsic rewards

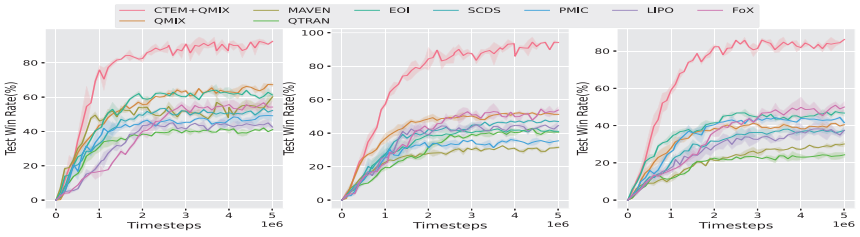
Figure 2: Performance comparison between our proposed CTEM and baselines in Pac-Men.



(a) 3s5z (easy) (b) 2c_vs_64zg (hard) (c)7sz (hard)

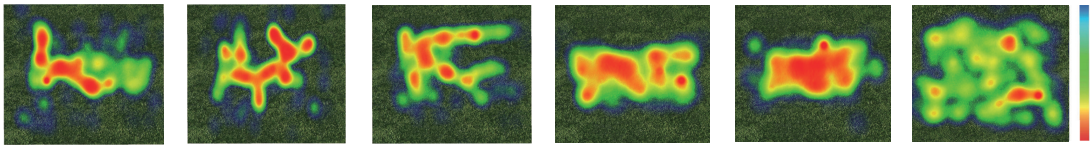
(d) 6h_vs_8z (super hard) (e) corridor (super hard) (f)3s5z_vs_3s6z (super hard)

Figure 3: Performance comparison between our proposed CTEM and baselines in the SMAC scenarios.



(a) terran_5_vs_5 (b) protoss_5_vs_5 (c)zerg_5_vs_5

Figure 4: Performance comparison between CTEM and baselines in the SMACv2 scenarios.



(a) MAVEN (b) EOI (c)SCDS (d) LIPO (e) FoX (f) CTEM

Figure 5: Visitation heatmaps of different algorithms in the terran_5_vs_5 scenario.

exploratory policies. We further provide the visitation heatmaps of agents trained by the baselines and our method in Figure 5. The results intuitively demonstrate that the movements of agents trained

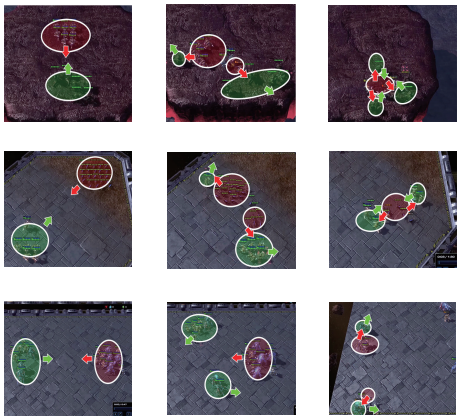


Figure 6: Visualization examples of diverse policies emerging in 6h_vs_8z (top), corridor (medium), and 3s5z_vs_3s6z (bottom) from initial (left) to final (right).

by the baselines are distributed in only partial areas of the environment. In contrast, our method motivates the agents to visit any possible states that contain environmental rewards and sufficiently explore the whole environment.

Scalability We next evaluate the scalability of our method with an increasing number of agents. As the number of agents increases, the state-action space expands exponentially, underscoring the critical need for exploration. In this section, we test the scalability of our method across four SMACv2 scenarios with an increasing number of agents: terran_5_vs_5, terran_10_vs_10, terran_15_vs_15, and terran_20_vs_20, with results shown in Table 2. Our method consistently outperforms QMIX in all scenarios. QMIX struggles to achieve satisfactory performance due to inadequate exploration. In contrast, our method exhibits robust scalability, demonstrating that maximizing the trajectory entropy allows for sufficient exploration.

Table 2: Performance of our method and QMIX in scenarios of SMACv2 with different number of agents

Method	terran_5_vs_5	terran_10_vs_10	terran_15_vs_15	terran_20_vs_20
CTEM+QMIX	0.82±0.03	0.84 ±0.03	0.81 ±0.03	0.78 ±0.04
QMIX	0.68±0.03	0.39±0.04	0.24 ±0.06	0.11±0.05

5.3 Ablation Study

In this section, we conduct several ablation studies to investigate the contribution of each component in our method. To investigate the impact of the autoregressive model used to learn the trajectory representation, we design a variant that ablates the autoregressive model and only uses the observation encoder. To measure the contribution of contrastive representation learning, we design a variant that encodes trajectories using a randomly initialized encoder with fixed parameters. Moreover, we also design a variant that learns the representations in a supervised manner by directly predicting the agent identities of trajectories. To test the entropy maximization objective, we design two variants that use the k-th nearest neighbor and randomly selected neighbors in the entropy, respectively.

We conduct experiments in 3 SMAC scenarios including 3s5z (easy), 2c_vs_64zg (hard), and corridor (super hard) to test these variants. The performance of ablation variants is shown in Figure 7a. Using the k-th nearest neighbor in the entropy estimator downgrades the performance and leads to a large variance. We also note an evident decrease in performance caused by using randomly selected neighbors. However, they still achieve higher win rates than QMIX, indicating our representation learning method is quite robust. As illustrated by Figure 7b, using k nearest neighbors in the entropy estimator allows for more efficient intrinsic rewards than the other two methods, encouraging sufficient exploration.

Using a fixed encoder without representation learning leads to sub-optimal performance, providing inefficient intrinsic rewards for agents to explore. Compared with self-supervised contrastive representation learning, the representations learned by predicting the agent identities of trajectories incur a noticeable decline in performance. We believe this is because the representations supervised by the agent identity are harmful to efficient exploration. Moreover, the ablation of the autoregressive model achieves similar performance to our method in 3s5z and 2c_vs_64zg, however, it yields a significant performance drop in the super hard corridor scenario. This phenomenon indicates that learning trajectory representations with the autoregressive model can lead to a more robust result, especially in challenging multi-agent tasks.

6 EVALUATIONS OF CTEM WITH DIFFERENT VALUES OF k

To test whether our method's performance is highly sensitive to k, we present its performance with various k values in the terran_5_vs_5 (with 5 agents) and terran_20_vs_20 (with 20 agents) scenarios, as shown in Table 3. The results indicate that different values of k result in only minor performance variations in both scenarios, demonstrating that our method remains robust across a range of k values.

7 COMPARISON WITH ϵ-GREEDY

The ϵ-greedy approach is a widely used exploration strategy in many reinforcement learning (RL) algorithms. Typically, increasing the ϵ value promotes more exploration. In this section, we compare our entropy maximization method to the ϵ-greedy strategy to demonstrate its advantages in encouraging exploration within MARL. For this comparison, we set the ϵ values to 0.05, 0.08, and 0.12 for QMIX and evaluate these settings across challenging scenarios, including corridor and 3s5z_vs_3s6z. The results, summarized in Table 4, indicate that our entropy maximization method is more effective at promoting exploration than simply increasing the ϵ values. Notably, higher ϵ values do not yield substantial performance improvements. In multi-agent environments, increasing ϵ primarily adds randomness to individual agents' action selection without improving coordination or diversity among agents, as it overlooks the trajectories of other agents. This leads to inefficient exploration.

8 RELATED WORKS

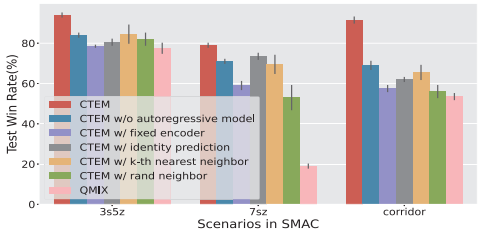
Agent Diversity Diversity in MARL settings aims to foster the learning of diverse policies among agents. For instance, SVO [20]

Table 3: Performance of our method with different values of k

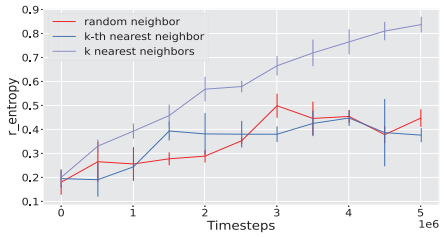
Method	terran_5_vs_5				terran_20_vs_20				
	k=1	k=2	k=3	k=4	k=1	k=4	k=10	k=15	k=18
CTEM+QMIX	0.87 ±0.07	0.84 ±0.03	0.86 ±0.05	0.90 ±0.02	0.76 ±0.09	0.72 ±0.04	0.77 ±0.03	0.75 ±0.03	0.78 ±0.04

Table 4: Comparisons of the performance of our method against QMIX with different values of ϵ

Method	corridor	3s5z_vs_3s6z
Trajectory entropy maximization (Ours)	0.92 ±0.03	0.87 ±0.04
ϵ = 0.05 (QMIX)	0.57 ±0.07	0.36 ±0.12
ϵ = 0.08 (QMIX)	0.63 ±0.05	0.41 ±0.08
ϵ = 0.12 (QMIX)	0.65 ±0.03	0.44 ±0.09



(a)



(b)

Figure 7: (a) Performance comparison of our method against different variants in the scenarios of SMAC. (b) Different kinds of intrinsic rewards in the corridor scenario.

draws upon social value orientation to tackle multi-agent social dilemmas. It achieves this by introducing an intrinsic reward that encourages agents to learn diverse policies. RODE [33] promotes diversity by assigning distinct actions to predefined roles. Although RODE is effective for agents with small action spaces, it may face challenges in scenarios with continuous actions and extensive action spaces. MAVEN [19] introduces a value-based approach that conditions joint behaviors of agents on a shared latent variable controlled by a hierarchical policy by maximizing the mutual information objective. EOI [10] learns a probabilistic classifier to predict the probability distribution over agents based on their observations. The correctly predicted probability serves as an intrinsic reward for policy training. CDS [14] focuses on encouraging multi-agent diversity by optimizing mutual information. It achieves this goal by creating lower bounds based on the Boltzmann softmax distribution and variational inference. PMIC [15] encourages learning of superior policies by maximizing the mutual information with regard to superior cooperative behaviors while minimizing mutual information associated with inferior behaviors. LIPO [2] regards policy compatibility as a proxy to learn diverse behaviors and identifies behaviors of each policy by maximizing the mutual information objective. FoX [12] introduces formation-based exploration, which promotes visiting diverse formations by directing agents to comprehensively understand their current formations. Despite their successes, they tend to overemphasize the relationship between the agent identity and trajectories. This emphasis sometimes leads agents to repeatedly visit similar observations, restricting their ability to explore new possibilities.

Entropy Maximization Entropy maximization has been employed in some RL works to efficiently encourage state exploration. RE3 [25] tries to improve sample efficiency by efficient exploration. It converts the high-dimensional observations into a compact low-dimensional representation space with a fixed encoder and leverages an entropy estimator to estimate state entropy in the low-dimensional representation space. Different from RE3, we adopt contrastive learning to learn a contrastive representation space, which captures more relevant information. APT [16] proposes a pre-training method maximzing the state entropy to explore the environment. An intrinsic reward based on the entropy estimator is used to train the agent policy in a reward-free environment. ProtoRL [37] learns representations through prototypes, which simultaneously serve a summary of the agent's exploration experience. The prototype based representations not only generalizes across tasks, but also efficiently accelerate exploration. In multi-agent settings, inspired by these methods, our method encourages multi-agent diversity by maximizing trajectory entropy in a contrastive representation space, which induces efficient exploration and collaboration.

9 CONCLUSION

Observing the behavioral similarities among agents arising from parameter sharing, in this paper, we propose a novel method encouraging multi-agent diversity through maximizing the entropy of agents' trajectories in an abstract contrastive representation space. We evaluate our method in multiple challenging benchmark tasks and demonstrate the significant outperformance of our method over existing state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported in part by the Fundamental Research Funds for the Central Universities (Grant No. NS2024055), in part by National Natural Science Foundation of China (62061146002), and in part by Natural Science Foundation of Jiangsu Province (Grant No. BK20222012).

REFERENCES

[1] Jan Beirlant, Edward J Dudewicz, László Györfi, Edward C Van der Meulen, et al. 1997. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences* 6, 1 (1997), 17–39.

[2] Rujikorn Charakorn, Poramate Manoonpong, and Nat Dilokthanakul. 2023. Generating Diverse Cooperative Agents by Learning Incompatible Policies. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=UkU05GOH7_6

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing*.

[5] Benjamin Ellis, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N Foerster, and Shimon Whiteson. 2022. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2212.07489* (2022).

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

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Siyi Hu, Chuanlong Xie, Xiaodan Liang, and Xiaojun Chang. 2022. Policy diagnosis via measuring role diversity in cooperative multi-agent rl. In *International Conference on Machine Learning*. PMLR, 9041–9071.

[9] Shariq Iqbal, Christian A Schroeder De Witt, Bei Peng, Wendelin Böhmer, Shimon Whiteson, and Fei Sha. 2021. Randomized entity-wise factorization for multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 4596–4606.

[10] Jiechuan Jiang and Zongqing Lu. 2021. The emergence of individuality. In *International Conference on Machine Learning*. PMLR, 4992–5001.

[11] Jiantao Jiao, Weihao Gao, and Yanjun Han. 2018. The nearest neighbor information estimator is adaptively near minimax rate-optimal. *Advances in neural information processing systems* 31 (2018).

[12] Yonghyeon Jo, Sunwoo Lee, Junghyuk Yeom, and Seungyul Han. 2024. FoX: Formation-aware exploration in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 12985–12994.

[13] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*. PMLR, 5639–5650.

[14] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. 2021. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 3991–4002.

[15] Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E Taylor, Wenyuan Tao, Zhen Wang, et al. 2022. PMIC: improving multi-agent reinforcement learning with progressive mutual information collaboration. *arXiv preprint arXiv:2203.08553* (2022).

[16] Hao Liu and Pieter Abbeel. 2021. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems* 34 (2021), 18459–18473.

[17] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[18] Xiaoteng Ma, Yiqin Yang, Chenghao Li, Yiwen Lu, Qianchuan Zhao, and Jun Yang. 2021. Modeling the Interaction between Agents in Cooperative Multi-Agent Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 853–861.

[19] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems* 32 (2019).

[20] Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes, and Joel Z Leibo. 2020. Social diversity and social preferences in mixed-motive reinforcement learning. *arXiv preprint arXiv:2002.02325* (2020).

[21] Kamal K Ndousse, Douglas Eck, Sergey Levine, and Natasha Jaques. 2021. Emergent social learning via multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 7991–8004.

[22] Frans A Oliehoek and Christopher Amato. 2015. A concise introduction to decentralized pomdps.

[23] T Rashid, CS De Witt, G Farquhar, J Foerster, S Whiteson, and M Samvelyan. 2018. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement Learning. In *35th International Conference on Machine Learning, ICML 2018*. 6846–6859.

[24] 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. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043* (2019).

[25] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2021. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*. PMLR, 9443–9454.

[26] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. 2003. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences* 23, 3-4 (2003), 301–321.

[27] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 5887–5896.

[28] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. 2021. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*. PMLR, 9870–9879.

[29] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 2085–2087.

[30] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[31] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2020. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062* (2020).

[32] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. 2020. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039* (2020).

[33] Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. 2020. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523* (2020).

[34] Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. 2020. Dop: Off-policy multi-agent decomposed policy gradients. In *International conference on learning representations*.

[35] Tong Wu, Pan Zhou, Kai Liu, Yali Yuan, Xiumin Wang, Huawei Huang, and Dapeng Oliver Wu. 2020. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology* 69, 8 (2020), 8243–8256.

[36] Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. 2021. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 10299–10312.

[37] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. 2021. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*. PMLR, 11920–11931.

[38] Tianhao Zhang, Yueheng Li, Chen Wang, Guangming Xie, and Zongqing Lu. 2021. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 12491–12500.