Learning Joint Behaviors with Large Variations

Tianxu Li, Kun Zhu*

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
{tianxuli, zhukun}@nuaa.edu.cn

Abstract

Cooperative Multi-Agent Reinforcement Learning (MARL) has drawn increasing interest in recent works due to its significant achievements. However, there are still some challenges impeding the learning of optimal cooperative policies, such as insufficient exploration. Prior works typically adopt mutual information-based methods to encourage exploration. However, this category of methods does not necessarily encourage agents to fully explore the joint behavior space. To address this limitation, we propose a novel objective based on learning a representation function with a Lipschitz constraint to maximize the traveled distances in the joint behavior space, encouraging agents to learn joint behaviors with large variations and leading to sufficient exploration. We further implement our method on top of QMIX. We demonstrate the effectiveness of our method by conducting experiments on the LBF, SMAC, and SMACv2 benchmarks. Our method outperforms previous methods in terms of final performance and state-action space exploration.

Introduction

Cooperative Multi-Agent Reinforcement Learning (MARL) has demonstrated its potential in addressing diverse multi-agent challenges, ranging from multiplayer video games (Vinyals et al. 2019) to real-world traffic light control (Wu et al. 2020). MARL facilitates effective cooperation by training multiple agents together to maximize team returns. Due to partial observation constraints and high scalability demands of multi-agent systems, Centralized Training with Decentralized Execution (CTDE) framework (Lowe et al. 2017), where decentralized policies are learned for agents to make action decisions while being trained in a centralized manner to ensure robust and stable performance, has been widely used in many MARL algorithms, such as value-decomposition methods (Iqbal et al. 2021; Yang et al. 2021; Wang et al. 2020a; Sunehag et al. 2018; Rashid et al. 2018) and policy gradients (Ma et al. 2021; Wang et al. 2020b; Ndousse et al. 2021; Zhang et al. 2021).

One of the most popular algorithms based on CTDE is QMIX (Rashid et al. 2018). QMIX learns factored value functions for each agent by optimizing an approximation for the joint action-value function. The factors allow agents to make action decisions only depending on their historical trajectories, enabling decentralized execution. Moreover, QMIX monotonically combines these factors with a mixing network and trains them with the global state and joint behaviors of all agents, leading to stable and robust performance. However, the trade-off for this decentralization is evident, given that the monotonicity constraint imposes limitations on QMIX, leading to suboptimal approximations of values and inefficient exploration (Mahajan et al. 2019). There have been many recent efforts trying to enhance the performance of QMIX by improving the network architecture (Sunehag et al. 2017; Rashid et al. 2018; Wang et al. 2020a; Son et al. 2019) or extensive exploration (Mahajan et al. 2019; Wang et al. 2019; Jiang and Lu 2021; Li et al. 2021, 2022; Yu et al. 2023).

MAVEN (Mahajan et al. 2019) realizes the committed exploration by enabling agents to condition their behaviors on the shared latent variables, controlled by a hierarchical policy, through maximizing mutual information between latent variables and joint behaviors. This leads the latent variable to be maximally informative of the joint behavior. Therefore, optimizing the mutual information objective encourages the visitations of mutually different joint behaviors. Given the promising results achieved by mutual information in facilitating exploration, many recent studies also adopt the mutual information objective to improve the performance of QMIX (Wang et al. 2019; Jiang and Lu 2021; Li et al. 2021, 2022; Yu et al. 2023).

Despite their achievements, these mutual information-based methods may not enable agents to sufficiently explore the joint behavior space. Since the mutual information remains unaffected by scaling or invertible transformations of input variables, it inherently possesses infinitely many optima (Ozair et al. 2019). Consequently, the convergence tends to occur towards the most easily optimizable maximum, i.e., the mutual information objective can be easily maximized with only slight differences between joint behaviors, which does not necessarily encourage agents to learn joint behaviors with large variations for sufficient exploration of the joint behavior space in complex multi-agent tasks.

In order to resolve the limitation of mutual information-based methods, we propose a novel multi-agent exploration

*Corresponding author.

method, called Direction-based Joint Behavior Difference (DJBD). The mutual information-based methods prefer joint behaviors that are closer in the joint behavior space rather than seeking for distant ones so that the maximum of mutual information can be easily achieved. To address this, our method is based on a novel direction item objective that aligns the directions of joint behavior differences in a latent representation space and latent variables, to learn a representation function with a Lipschitz constraint. The Lipschitz constraint ensures that the maximization of our objective in the latent space can lead to increased traveled distances[1] in the joint behavior space. Compared to previous mutual information-based methods, our method causes large traveled distances in the joint behavior space, thus leading to more sufficient exploration and the learning of complex cooperative behaviors.

The contributions of this work can be summarized as follows:

- We propose a novel multi-agent exploration method named DJBD to learn a representation function with a Lipschitz constraint by maximizing a direction term objective, enlarging the traveled distances in the joint behavior space. We further deploy it in QMIX by introducing a per-step transition reward function for agents to encourage efficient exploration.
- We conduct experiments on the Level-Based Foraging (LBF) (Albrecht and Stone 2019), StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019), and SMACv2 (Ellis et al. 2022) benchmarks to test our proposed DJBD. The experimental results demonstrate the superior performance of our method compared to existing state-of-the-art methods. Our method is more robust in learning complex cooperative policies in challenging multi-agent tasks.

Related Works

Collaborative MARL Numerous efforts have been dedicated to enhancing agent performance in collaborative environments within the MARL field. Policy-based methods like COMA (Foerster et al. 2018), MADDPG (Lowe et al. 2017), MAPPO (Yu et al. 2022), and DOP (Wang et al. 2020b) learn decentralized policies for agents to maximize expected team returns by employing the CTDE framework. On the other hand, value-based algorithms like VDN (Sunehag et al. 2017), QMIX (Rashid et al. 2018), and QPLEX (Wang et al. 2020a) learns an approximation for the joint action value function and decompose it into individual utility functions, enabling agents to make decentralized action decisions. However, these methods face challenges in learning optimal cooperative policies maximizing the joint action-value function due to representational constraint, potentially hindering sufficient exploration. While VDN uses a summation of agent utilities, QMIX introduces a mixing network with a monotonic constraint to relax the summation constraint, and QPLEX further employs a duplex dueling network architecture for enhanced representation capability. Despite their large representation capability, agents still suffers from insufficient exploration to learn optimal cooperative policies in complex multi-agent tasks.

Mutual Information-based Cooperation Several recent works adopt the mutual information objective to improve exploration. MAVEN (Mahajan et al. 2019) introduces a latent variable controlled by a hierarchical policy for value-based agents to condition on. MAVEN maximizes mutual information between the latent variable and trajectories to learn diverse joint behaviors. EOI (Jiang and Lu 2021) encourages individuality by training a probabilistic classifier to predict the distribution over agents based on observations, using this prediction as an intrinsic reward for policy training. CDS (Li et al. 2021) optimizes mutual information through lower bounds derived from Boltzmann softmax distribution and variational inference, introducing intrinsic rewards to address mutual information objectives. Similarly, EITI (Wang et al. 2019) uses mutual information to extract transition dynamics, while PMIC (Li et al. 2022) maximizes mutual information for superior cooperative behaviors while minimizing it for inferior ones. GHQ (Yu et al. 2023) learns individual parameters for agent groups through Inter-group Mutual Information (IGMI) to encourage diversity. CIA (Liu et al. 2023) achieves temporary credits distinguishability among agents by maximizing the mutual information between temporary credits and identity representations. LIPO (Charakorn, Manoonpong, and Dilokthanakul 2023) employs policy compatibility as a means to learn diverse policies, further diversifying agents' behaviors via the mutual information objective. FoX (Jo et al. 2024) introduces formation-based exploration, encouraging diverse formations by guiding agents to be aware of their current formations via maximizing the mutual information between formations and latent variables. These methods exhibit promise in promoting exploration, yet they may prefer similar joint behaviors or trajectories that can easily achieve the maximum of the mutual information, hindering sufficient exploration.

Backgrounds

Multi-Agent System

We consider a fully cooperative multi-agent Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek and Amato 2015), denoted as $\langle A, S, U, P, R, O, \Omega, \gamma \rangle$. Here, A denotes a set of $|A|$ agents, $s \in S$ is the global state, and U represents the set of actions. Each time step initiates with each agent a receiving an observation $o^a \in \Omega$ via the function $O(s, a)$. Subsequently, the agent selects an action $u^a \in U$. The collection of all agents' behaviors forms a joint behavior $\boldsymbol{u}$; the environment transitions to the next state s' based on the probability from the transition function $P\left(s' \mid s, \boldsymbol{u}\right)$. Simultaneously, the environment provides agents with a shared team reward $r = R(s, \boldsymbol{u})$. $\gamma \in [0, 1)$ is the discount factor. The observation-action pairs $\langle o^a, u^a \rangle$ for agent a form its trajectory $\tau^a \in \mathcal{T}$. Each agent a learns an individual policy $\pi^a\left(u^a \mid \tau^a\right)$, collectively forming a joint policy $\boldsymbol{\pi}$

[1]The traveled distances refer to the dot product of the joint behavior distances and the latent variables.

to maximize the joint action-value function $Q^{\boldsymbol{\pi}}(s, \boldsymbol{u}) = \mathbb{E}_{s_{0:\infty}, \boldsymbol{u}_{0:\infty}} [\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, \boldsymbol{u}_0 = \boldsymbol{u}, \boldsymbol{\pi}]$.

Direction-based Joint Behavior Difference

In this section, we propose Direction-based Joint Behavior Difference (DJBD), a novel multi-agent exploration method encouraging the visitations of diverse joint behaviors with larger traveled distances. We first analyze the limitation of mutual information in encouraging diverse joint behaviors. Then, we derive our objective from the mutual information and learn a representation function by maximizing our objective with the 1-Lipschitz constraint. Further, we provide a practical learning algorithm by integrating our method with QMIX.

Limitation of mutual information-based methods We first discuss why previous mutual information-based methods (Mahajan et al. 2019; Li et al. 2022; Charakorn, Manoonpong, and Dilokthanakul 2023; Jo et al. 2024) may lead to insufficient exploration. The mutual information objective between the joint behavior $\boldsymbol{u_t}$ and the latent variable z with a variational lower bound can be written as follows:

$$\begin{aligned} I(z; \boldsymbol{u_t}) &= \mathbb{E}_{z, \boldsymbol{u_t}} [\log p(z \mid \boldsymbol{u_t})] + \mathbb{E}_z [\log p(z)] \\ &\geq \mathbb{E}_{z, \boldsymbol{u_t}} [\log q(z \mid \boldsymbol{u_t})] + \mathbb{E}_z [\log p(z)] \end{aligned} \quad (1)$$

where $\mathbb{E}_z [\log p(z))]$ is a constant since the latent variable $z \in \mathbb{R}^d$ is sampled from a fixed uniform distribution $p(z)$. The variational distribution $q(z \mid \boldsymbol{u_t})$ is an approximation of $p(z \mid \boldsymbol{u_t})$, the true posterior of z. The optimization over the mutual information means optimizing over the variational distribution $q(z \mid \boldsymbol{u_t})$.

Although the objective of mutual information shown in Equation 1 encourages mutually diverse joint behaviors given different values of z, one limitation is that the goal of fully maximizing mutual information can be achieved even with small differences between the joint behaviors. This is because the variational distribution $q(z \mid \boldsymbol{u_t})$ serves as a joint behavior discriminator that can be easily modeled when different values of z correspond to marginally distinct joint behaviors $\boldsymbol{u_t}$, which means that the mutual information objective lacks a metric for measuring the distance in the joint behavior space. This limitation becomes extremely problematic for many challenging cooperative tasks because small differences between joint behaviors may impede sufficient exploration of the joint behavior space, which leads to sub-optimal cooperative policies and downgrade the performance of agents.

Direction-based Joint Behavior Difference To resolve this limitation and maximize traveled distances between joint behaviors, we adopt neither mutual information maximization nor a joint behavior discriminator but a representation learning function that learns the latent representations of joint behaviors. Concretely, to derive our objective, consider the variational mutual information between joint behaviors and the latent variable z with a conditional form (Sharma et al. 2019),

$$\begin{aligned} I(z; \boldsymbol{u_{t+1}} \mid \boldsymbol{u_t}) &\geq \mathbb{E}_{z, \boldsymbol{u_{t+1}}} [\log q(z \mid \boldsymbol{u_t}, \boldsymbol{u_{t+1}})] \\ &\quad + \mathbb{E}_z [\log p(z \mid \boldsymbol{u_t})] \\ &= -\frac{1}{2} \mathbb{E}_{z, \boldsymbol{u_{t+1}}} \left[\| z - \mu(\boldsymbol{u_t}, \boldsymbol{u_{t+1}}) \|^2 \right] \\ &\quad - \frac{d}{2} \log(2\pi) + \mathbb{E}_z [\log p(z \mid \boldsymbol{u_t}))] . \end{aligned} \quad (2)$$

We further parameterize $q(z \mid \boldsymbol{u_t}, \boldsymbol{u_{t+1}})$ by a normal distribution with unit variance $\mathcal{N}(\mu(\boldsymbol{u_t}, \boldsymbol{u_{t+1}}), \mathbf{I})$. Here we assume $p(z \mid \boldsymbol{u_t}) = p(z)$. d is the dimensionality of z. Maximizing $I(z; \boldsymbol{u_{t+1}} \mid \boldsymbol{u_t})$ encourages different values of z to correspond to distinct $\boldsymbol{u_{t+1}}$ with the last step joint behavior being $\boldsymbol{u_t}$.

The intuition behind our introduction of the conditional mutual information objective $I(z; \boldsymbol{u_{t+1}} \mid \boldsymbol{u_t})$ is that we need to highlight the differences between joint behaviors induced by the policies of agents at successive time steps. To achieve this goal, we further model $\mu(\boldsymbol{u_t}, \boldsymbol{u_{t+1}})$ with $\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})$. Here, ϕ is a learnable representation function that maps the joint behaviors to an abstract representation space. With the representation function, we can decompose the variational distribution $q(z \mid \boldsymbol{u_t}, \boldsymbol{u_{t+1}})$ as follows,

$$\begin{aligned} &\mathbb{E}_{z, \boldsymbol{u_{t+1}}} [\log q(z \mid \boldsymbol{u_t}, \boldsymbol{u_{t+1}})] \\ &= -\frac{1}{2} \mathbb{E}_{z, \boldsymbol{u_{t+1}}} \left[\| z - \mu(\boldsymbol{u_t}, \boldsymbol{u_{t+1}}) \|^2 \right] - \frac{d}{2} \log(2\pi) \\ &= -\frac{1}{2} \mathbb{E}_{z, \boldsymbol{u_{t+1}}} \left[\| z - (\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})) \|^2 \right] - \frac{d}{2} \log(2\pi) \\ &= \mathbb{E}_{z, \boldsymbol{u_{t+1}}} \left[(\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t}))^\top z \right] \\ &- \frac{1}{2} \mathbb{E}_{z, \boldsymbol{u_{t+1}}} \left[\| \phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t}) \|^2 \right] \\ &- \frac{1}{2} \mathbb{E}_z \left[z^\top z \right] - \frac{d}{2} \log(2\pi) \end{aligned} \quad (3)$$

where $\mathbb{E}_z \left[z^\top z \right]$ is a constant because $p(z)$ is a fixed distribution. The first term of Equation 3 aligns the directions of the vector $\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})$ and latent variables z, while the second term serves as a norm regularizer.

To encourage diverse joint behaviors with larger traveled distances, we use the direction term of Equation 3 to be the objective of our method to train the representation function ϕ, which can be written as follows:

$$J_d = \mathbb{E}_{z, \boldsymbol{u_{t+1}}} \left[(\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t}))^\top z \right] . \quad (4)$$

The $\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})$ term guarantees the distance between joint behaviors in the latent representation space. However, simply maximizing the objective J_d may cause the value of $\phi(\boldsymbol{u_{t+1}})$ to be infinite regardless of the real joint behavior $\boldsymbol{u_{t+1}}$, i.e., enlarging the distance in the representation space may not lead to increased distance in the joint behavior space. To address this issue, we introduce the 1-Lipschitz constraint and enforce it on the representation

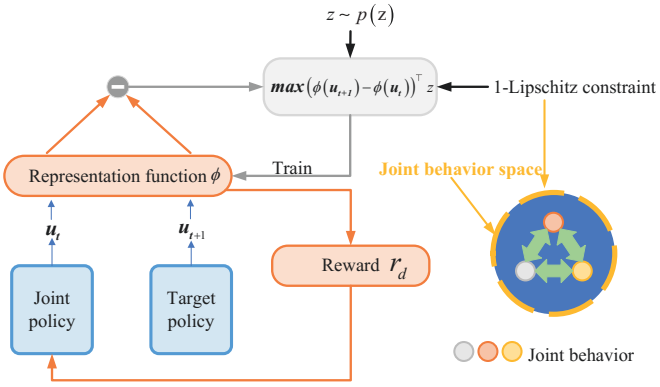


Figure 1: Architecture for DJBD.

function ϕ to ensure that the increased distance in the representation space entails an increase in the distances between joint behaviors. The Lipschitz-constrained objective can be written as follows,

$$J_d = \mathbb{E}_{z,\boldsymbol{u_{t+1}}}\left[\left(\phi\left(\boldsymbol{u_{t+1}}\right) - \phi\left(\boldsymbol{u_t}\right)\right)^\top z\right] \\ \text{s.t.} \quad \forall x, y \in \boldsymbol{U} \quad \|\phi(x) - \phi(y)\| \leq \|x - y\| \tag{5}$$

We note that maximizing the objective in Equation 5 increases the length of $\phi\left(\boldsymbol{u_{t+1}}\right) - \phi\left(\boldsymbol{u_t}\right)$. Meanwhile, it also leads to an increase in the upper bound $\|\boldsymbol{u_{t+1}} - \boldsymbol{u_t}\|$ caused by the 1-Lipschitz constraint (i.e., $\|\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})\| \leq \|\boldsymbol{u_{t+1}} - \boldsymbol{u_t}\|$). Therefore, the maximization of our objective shown in Equation 5 encourages agents to take joint behaviors with larger traveled distances. In practice, we implement the 1-Lipschitz constraint by Spectral Normalization (Miyato et al. 2018).

The 1-Lipschitz constraint used in our method is entirely different from its usage in unsupervised skill discovery to learn useful skills (Choi et al. 2024; Yang et al. 2023; Mazzaglia et al. 2022; Park et al. 2021). First, we derive our objective from a novel conditional mutual information that associates the next step joint behaviors with latent variables given the current joint behaviors. Second, we impose the 1-Lipschitz constraint between joint behaviors in consecutive time steps instead of the initial and final states, which allows for efficient per-step incentives.

Practical Learning Algorithm Since we have presented the objective of our method, we next provide a practical learning algorithm that combines our method with QMIX (Rashid et al. 2018), a MARL algorithm based on the value-decomposition that jointly optimizes Q_{tot}, an estimate of the joint action-value function, to learn optimal policies for agents. To implement our method in MARL, we introduce a per-step transition reward function for agents given as:

$$r_d(s_t, \boldsymbol{u_t}, s_{t+1}) = (\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t}))^\top z \tag{6}$$

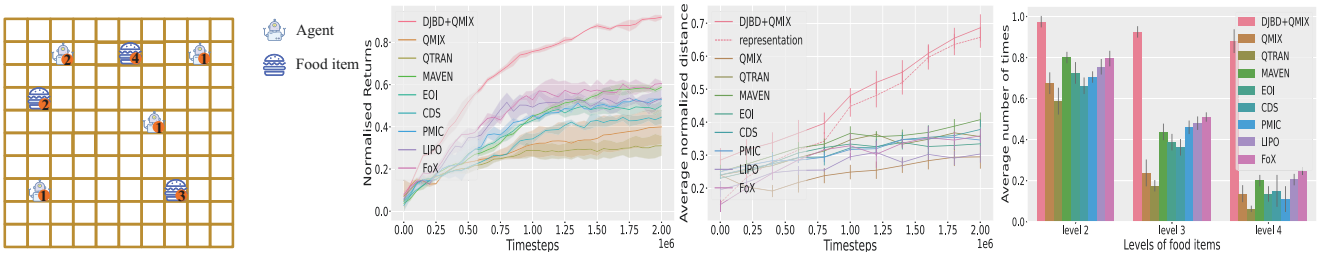
where the transition $(s_t, \boldsymbol{u_t}, s_{t+1})$ is sampled from the replay buffer and $\boldsymbol{u_{t+1}}$ is selected by the target utility networks $\hat{Q}^a$ in QMIX. We use the per-agent utilities $Q^a(\tau^a, u^a)$ in the case of QMIX as the representatives of joint behaviors to be the input of the representation function ϕ. To learn diverse joint behaviors with larger traveled distances, we apply this reward function to the TD loss of QMIX to train the policies of agents. The TD loss aims to update the Q_{tot} network of QMIX towards maximizing expected team returns that is given as:

$$\mathcal{L}_{TD} = \sum_{i=1}^{b}\left[\left(y^{\text{tot}} - Q_{tot}(\boldsymbol{\tau_t}, \boldsymbol{u_t}, s_t)\right)^2\right], \text{where} \\ y^{tot} = r + \alpha r_d + \gamma \max_{\boldsymbol{u_{t+1}}} \hat{Q}_{tot}\left(\boldsymbol{\tau_{t+1}}, \boldsymbol{u_{t+1}}, s_{t+1}\right) \tag{7}$$

where b is the batch size of the transitions sampled from the replay buffer, r is the environmental reward, α is a coefficient that is used to weight the transition reward r_d, and $\hat{Q}_{tot}$ is the target network. We provide the pseudocode for our method in the Appendix 3. Note that our method is easy to implement since it only additionally introduces a transition reward function based on the representation function ϕ to QMIX, and the other components are the same as QMIX. In order to achieve large joint behavior variations and maximize team rewards, we train the representation function ϕ and Q_{tot} alternatively. Moreover, we also integrate our method with the policy gradient method MAPPO. We refer the reader to Appendix 1 for more details.

Experiments

In this section, we evaluate the performance of our proposed DJBD on challenging multi-agent tasks from the LBF (Albrecht and Stone 2019), SMAC (Samvelyan et al. 2019), and SMACv2 (Ellis et al. 2022) benchmarks to highlight its superior performance. We use several state-of-the-art methods as baselines, including value-decomposition methods such as QMIX (Rashid et al. 2018) and QTRAN (Son et al. 2019), as well as mutual information-based exploration methods such as MAVEN (Mahajan et al. 2019), EOI (Jiang and Lu 2021), CDS (Li et al. 2021), PMIC (Li et al. 2022), LIPO (Charakorn, Manoonpong, and Dilokthanakul 2023), and FoX (Jo et al. 2024). The experimental results presented in this paper are illustrated with both the mean and standard



(a) Illustration of LBF (b) Learning curves (c) Average normalized distance $\|\boldsymbol{u}_{t+1} - \boldsymbol{u}_t\|$ (d) Average number of times the food items are eaten

Figure 2: Performance comparison between our proposed DJBD and baselines in LBF.

deviation of performance tested with five random seeds. For a fair comparison, the hyperparameters shared across methods are consistent in each multi-agent task. Training details and hyperparameters used in our experiments are available in the Appendix 5.

Level-Based Foraging

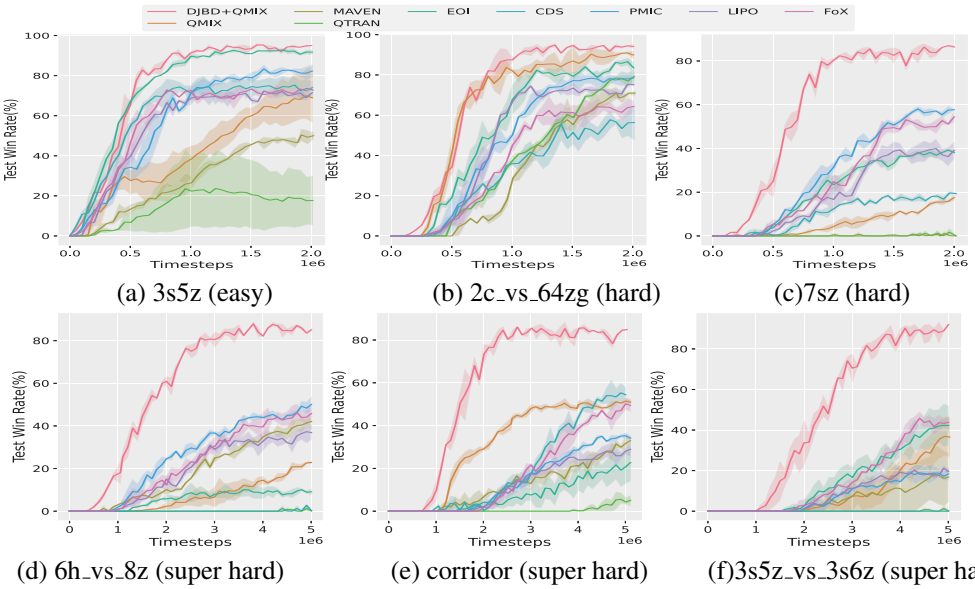
To demonstrate the effectiveness of our proposed DJBD in promoting multi-agent exploration, we first introduce the Level-Based Foraging (LBF) environment (Albrecht and Stone 2019), where several agents and food items are randomly distributed in a 10x10 grid world as shown in Figure 2a. Each agent can only observe a 3x3 grid around them. During each episode, agents can move in four directions in the grid world and collect items adjacent to them. Each agent and item is assigned a specific level. The item can be successfully collected only if the total level of agents attempting to collect it is equal to or greater than its level. This challenging collection task in the LBF environment requires agents to collaborate with each other to attain more rewards, necessitating the emergence of complex cooperative policies.

The performance of different algorithms in the LBF environment is shown in Figure 2b. Our proposed DJBD achieves superior performance compared to the baselines. We visualize the average normalized distance between joint behaviors $\|\boldsymbol{u}_{t+1} - \boldsymbol{u}_t\|$ in Figure 2c. The results demonstrate that our method efficiently enlarges the distance between joint behaviors by maximizing the direction term objective. Also, the distance between joint behaviors increases as the distance in the representation space increases, due to the 1-Lipschitz constraint. In contrast to our method, the mutual information-based baselines such as EOI, CDS, and FoX prefer joint behaviors with small distances. We believe this is because such slight differences between joint behaviors can be sufficient to fully maximize the mutual information objective, thus these baselines have no incentives to seek distant joint behaviors, which leads to insufficient exploration and downgrades the final performance in the LBF environment. Notably, QMIX fails to yield satisfactory final performance due to limited exploration. But with the help of our method, QMIX successfully learns exploratory policies for agents to achieve more rewards. We further provide the average number of times food items are eaten by agents in Figure 2d. The baselines struggle to learn effective collaborative policies to collect the food items, especially the food item with level 4 requiring at least 3 agents to collaboratively collect. Compared with baselines, DJBD yields robust results and maintains its outperformance across all levels of food items, demonstrating that our method learns more complex cooperative policies via enlarging the traveled distance in joint behavior space.

SMAC

We next evaluate our proposed DJBD in the StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019), a common-used benchmark for evaluating cooperative MARL algorithms. It includes a set of challenging combat scenarios that require agents to learn micromanagement skills to defeat enemies controlled by built-in AI. These scenarios are classified into different difficulty levels (easy, hard, and super hard). To validate the effectiveness of our method, we compare our method with baselines 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). The performance is not comparable between different versions of SMAC. The version of SMAC we use in our work is SC2.4.10.

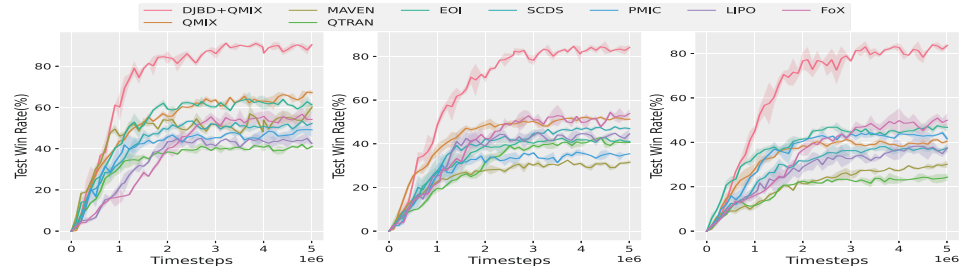
The performance of our method and baselines is shown in Figure 3. Our method achieves better performance than baselines. The baseline QMIX achieves satisfactory performance in the easy 3s5z scenario and the hard 2c_vs_64zg scenario, however, it fails to learn optimal policies in the other more challenging scenarios and needs our method to improve its final performance. EOI is less efficient in learning cooperative policies since its observation classifier may overfit agent identity information. Our method does not rely on the agent identity but on the differences between joint behaviors, thus leading to more efficient exploration. Compared with MAVEN, our method maintains its outperformance in the super hard scenarios by enlarging the traveled distances between joint behaviors for exploring cooperative policies. Moreover, our method does not yield significant improvement over baselines in the easy 3s5z scenario and the hard 2c_vs_64zg scenario, demonstrating that encouraging the differences between joint behaviors may not result in better learning efficiency in simple scenarios that do not



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

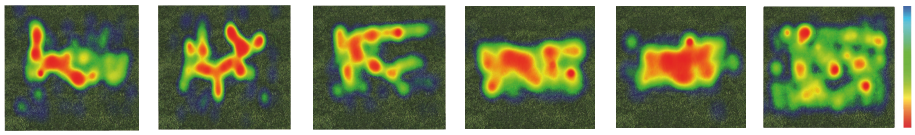
(d) 6h_vs_8z (super hard) (e) corridor (super hard) (f)3s5z_vs_3s6z (super hard)

Figure 3: Performance comparison between our proposed DJBD 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 our method DJBD and baselines in the SMACv2 scenarios.



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

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

require complex tactics.

Exploration and Stochasticity Although the scenarios in SMAC are challenging, however, the agent compositions and initial positions during each episode are the same in each scenario. This means that it lacks enough stochasticity to evaluate the exploration of MARL algorithms. The agents may learn fixed action sequences regardless of specific observations. Therefore, to evaluate the state-action space exploration, we further use a more challenging benchmark called SMACv2 (Ellis et al. 2022), which adds randomness by adopting random unit types and random initial positions.

Our proposed DJBD achieves significant outperformance in all scenarios, as shown in Figure 4. MAVEN does not yield satisfactory performance in most scenarios and even achieves worse performance than QMIX. We believe this is because the maximization of mutual information does not lead to sufficient exploration to adapt to the changes in the environment caused by stochasticity. Taking advantage of learning the Lipschitz-constrained representation function, our method successfully learns more exploratory policies and achieves higher win rates. Compared to EOI, CDS, and FoX which encourage multi-agent diversity, our method performs substantially better than them, indicating enlarging traveled distances between joint behaviors is more efficient in promoting exploration.

To demonstrate the effectiveness of DJBD in encouraging

exploration, we further present visitation heatmaps for different methods in Figure 5. The mutual information-based baselines do not fully explore the map. The agent movements are only located in partial areas of the map. Therefore, the agents struggle to detect and cooperatively defeat distant enemies. However, by enlarging the traveled distance between joint behaviors of agents, as shown in Figure 5, the agents trained by our method actively explore the map and efficiently search for enemies. The agent movements are uniformly distributed on the map.

Ablations and Visualization

Our proposed DJBD consists of three main components: (i) the 1-Lipschitz constraint, (ii) the direction term, and (iii) the differences $\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})$. We carry out ablation studies to test their contributions. To test the 1-Lipschitz constraint, we design a variant that ablates the 1-Lipschitz constraint and only uses the direction term as our objective. To test the direction term, we use the normal distribution given by Equation 3 as our objective in place of the direction term. Moreover, we also design a variant using the regularizer term given by Equation 3 as our objective. To test the differences $\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})$, we ablate the differences by using $\phi(\boldsymbol{u_t})$, $\phi(\boldsymbol{u_{t+1}})$, and $\phi(\boldsymbol{u_{t+1}} - \boldsymbol{u_t})$, respectively.

We first conduct ablation studies in the scenarios of SMAC, and the results are shown in Figure 6. We note that missing any of the components constituting DJBD leads to a dramatic performance decrease. All the components contribute to the large joint behavior variations. Specifically, without the 1-Lipschitz constraint, the performance of agents declines rapidly and with similar performance to QMIX, indicating that directly maximizing the distance in the representation space would not lead to increased traveled distance in the real joint behavior space. In contrast, with the 1-Lipschitz constraint, our method consistently achieves robust performance in all scenarios. Maximizing our objective with the 1-Lipschitz constraint entails increased distance in the joint behavior space. The normal distribution does not improve the performance since it does not necessarily encourage large joint behavior variations as slight differences between joint behaviors are enough to fully maximize the variational objective. Simply maximizing the differences $\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})$ with the 1-Lipschitz constraint also results in sub-optimal performance. This is because the direction term objective enables the differences $\phi(\boldsymbol{u_{t+1}}) - \phi(\boldsymbol{u_t})$ to collapse to different directions given the latent variables, thus resulting in better joint behavior space converge. Moreover, using the representations of the joint behavior $\boldsymbol{u_t}$ and $\boldsymbol{u_{t+1}}$ in our objective only aligns the directions of representations of joint behaviors and latent variables, and does not actually enlarge the traveled distance between joint behaviors. Similarly, we note that learning the representations of the joint behavior difference $\phi(\boldsymbol{u_{t+1}} - \boldsymbol{u_t})$ also leads to poor performance. We believe this is because $\phi(\boldsymbol{u_{t+1}} - \boldsymbol{u_t})$ does not induce differences in the representation space to ensure the 1-Lipschitz constraint works properly.

We further visualize the final policies learned by our method in the corridor scenario, presented in Figure 7. Facing powerful enemies, the agents trained by our method cooperatively distribute the enemies' attacks. Specifically, one agent moves away from the team to attract the attention of the majority of enemies. This agent continuously leads the enemies away, diverting their fire to provide cover for the rest of the team. Meanwhile, the other agents strategically move in different directions and surround the following enemies. They can easily defeat the few remaining enemies, as most are attracted by the separated agent. As a result, the strength of enemies is greatly weakened. Enlarging the traveled distance in the joint behavior space allows agents to efficiently learn complex cooperative policies.

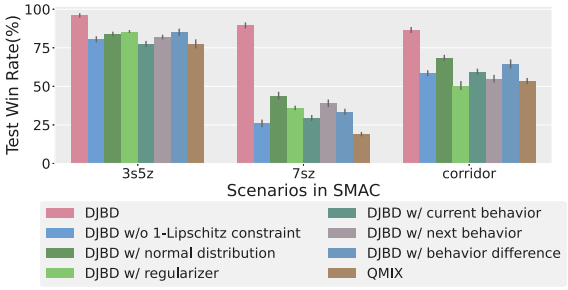


Figure 6: Performance comparison between DJBD and ablation variants in SMAC scenarios.

Figure 7: Visualization examples of agent policies learned by DJBD in the corridor scenario. Green and red shadows represent agents and enemies, respectively. Green and red arrows represent the moving directions of agents and enemies, respectively.

Conclusion

In this paper, we present a novel multi-agent exploration method called DJBD to learn optimal cooperative policies in complex multi-agent tasks. We analyze the limitation of mutual information-based methods that the agents may prefer similar joint behaviors that can easily maximize the mutual information objective, impeding sufficient exploration. We thus propose a novel objective based on learning a representation function with a Lipschitz constraint to resolve this limitation. Our method efficiently enlarges the traveled distance between joint behaviors, thus resulting in more diverse joint behaviors and sufficient exploration. We implement our method on top of QMIX and evaluate its performance in the LBF, SMAC, and SMACv2 benchmarks. Our method achieves better final performance and state-action space exploration than previous methods.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (62061146002), and in part by Natural Science Foundation of Jiangsu Province (Grant No. BK20211567, BK20222012).

References

Albrecht, S. V.; and Stone, P. 2019. Reasoning about hypothetical agent behaviours and their parameters. *arXiv preprint arXiv:1906.11064*.

Charakorn, R.; Manoonpong, P.; and Dilokthanakul, N. 2023. Generating Diverse Cooperative Agents by Learning Incompatible Policies. In *The Eleventh International Conference on Learning Representations*.

Choi, J.; Lee, S.; Wang, X.; Sohn, S.; and Lee, H. 2024. Unsupervised Object Interaction Learning with Counterfactual Dynamics Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11570–11578.

Ellis, B.; Moalla, S.; Samvelyan, M.; Sun, M.; Mahajan, A.; Foerster, J. N.; and Whiteson, S. 2022. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2212.07489*.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Iqbal, S.; De Witt, C. A. S.; Peng, B.; Böhmer, W.; Whiteson, S.; and Sha, F. 2021. Randomized entity-wise factorization for multi-agent reinforcement learning. In *International Conference on Machine Learning*, 4596–4606. PMLR.

Jiang, J.; and Lu, Z. 2021. The emergence of individuality. In *International Conference on Machine Learning*, 4992–5001. PMLR.

Jo, Y.; Lee, S.; Yeom, J.; and Han, S. 2024. FoX: Formation-aware exploration in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12985–12994.

Li, C.; Wang, T.; Wu, C.; Zhao, Q.; Yang, J.; and Zhang, C. 2021. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 3991–4002.

Li, P.; Tang, H.; Yang, T.; Hao, X.; Sang, T.; Zheng, Y.; Hao, J.; Taylor, M. E.; Tao, W.; Wang, Z.; et al. 2022. PMIC: improving multi-agent reinforcement learning with progressive mutual information collaboration. *arXiv preprint arXiv:2203.08553*.

Liu, S.; Zhou, Y.; Song, J.; Zheng, T.; Chen, K.; Zhu, T.; Feng, Z.; and Song, M. 2023. Contrastive identity-aware learning for multi-agent value decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11595–11603.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Ma, X.; Yang, Y.; Li, C.; Lu, Y.; Zhao, Q.; and Yang, J. 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.

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

Mazzaglia, P.; Verbelen, T.; Dhoedt, B.; Lacoste, A.; and Rajeswar, S. 2022. Choreographer: Learning and adapting skills in imagination. *arXiv preprint arXiv:2211.13350*.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Ndousse, K. K.; Eck, D.; Levine, S.; and Jaques, N. 2021. Emergent social learning via multi-agent reinforcement learning. In *International conference on machine learning*, 7991–8004. PMLR.

Oliehoek, F. A.; and Amato, C. 2015. A concise introduction to decentralized pomdps.

Ozair, S.; Lynch, C.; Bengio, Y.; Van den Oord, A.; Levine, S.; and Sermanet, P. 2019. Wasserstein dependency measure for representation learning. *Advances in Neural Information Processing Systems*, 32.

Park, S.; Choi, J.; Kim, J.; Lee, H.; and Kim, G. 2021. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*.

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

Samvelyan, M.; Rashid, T.; De Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2019. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, 5887–5896. PMLR.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; 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.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2020a. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*.

Wang, T.; Wang, J.; Wu, Y.; and Zhang, C. 2019. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*.

Wang, Y.; Han, B.; Wang, T.; Dong, H.; and Zhang, C. 2020b. Dop: Off-policy multi-agent decomposed policy gradients. In *International conference on learning representations*.

Wu, T.; Zhou, P.; Liu, K.; Yuan, Y.; Wang, X.; Huang, H.; and Wu, D. O. 2020. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(8): 8243–8256.

Yang, R.; Bai, C.; Guo, H.; Li, S.; Zhao, B.; Wang, Z.; Liu, P.; and Li, X. 2023. Behavior contrastive learning for unsupervised skill discovery. In *International Conference on Machine Learning*, 39183–39204. PMLR.

Yang, Y.; Ma, X.; Li, C.; Zheng, Z.; Zhang, Q.; Huang, G.; Yang, J.; and Zhao, Q. 2021. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 10299–10312.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35: 24611–24624.

Yu, X.; Lin, Y.; Wang, X.; Han, S.; and Lv, K. 2023. GHQ: Grouped Hybrid Q Learning for Heterogeneous Cooperative Multi-agent Reinforcement Learning. *arXiv preprint arXiv:2303.01070*.

Zhang, T.; Li, Y.; Wang, C.; Xie, G.; and Lu, Z. 2021. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. In *International Conference on Machine Learning*, 12491–12500. PMLR.