Encouraging metric-aware diversity in contrastive representation space

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

Abstract

In cooperative Multi-Agent Reinforcement Learning (MARL), agents that share policy network parameters often learn similar behaviors, which hinders effective exploration and can lead to suboptimal cooperative policies. Recent advances have attempted to promote multi-agent diversity by leveraging the Wasserstein distance to increase policy differences. However, these methods cannot effectively encourage diverse policies due to ineffective Wasserstein distance caused by the policy similarity. To address this limitation, we propose Wasserstein Contrastive Diversity (WCD) exploration, a novel approach that promotes multi-agent diversity by maximizing the Wasserstein distance between the trajectory distributions of different agents in a latent representation space. To make the Wasserstein distance meaningful, we propose a novel next-step prediction method based on Contrastive Predictive Coding (CPC) to learn distinguishable trajectory representations. Additionally, we introduce an optimized kernel-based method to compute the Wasserstein distance more efficiently. Since the Wasserstein distance is inherently defined for two distributions, we extend it to support multiple agents, enabling diverse policy learning. Empirical evaluations across a variety of challenging multi-agent tasks demonstrate that WCD outperforms existing state-of-the-art methods, delivering superior performance and enhanced exploration.

1 Introduction

Multi-Agent Reinforcement Learning (MARL) has shown promise in addressing various multi-agent challenges, such as multiplayer video games [Vinyals et al., 2019] and autonomous cars [Cao et al., 2012], attracting growing interest in recent years. MARL facilitates efficient collaboration by training multiple agents together towards maximizing team rewards. Yet, there are still many challenges such as partial observation constraints and high scalability requirements, when learning effective cooperative policies for agents in complex multi-agent tasks. To resolve these issues, recent works commonly employ the Centralized Training with Decentralized Execution (CTDE) framework [Lowe et al., 2017] where agents make decisions based on local observations using a decentralized policy jointly trained with global information, ensuring robust and stable performance.

The CTDE framework develops distinct decentralized policies for each agent, but training numerous policy network parameters can be inefficient. Thus, parameter sharing has become universal, allowing agents to share the same policy network parameters for action decision-making. This practice significantly reduces the number of parameters, leading to lower computational cost and speeding up training. Additionally, parameter sharing promotes experience sharing during centralized training, fostering robust policy learning and improving overall efficiency [Wang et al., 2020b]. Despite these benefits, sharing policy parameters can lead to homogeneous behaviors among agents, hindering multi-agent diversity and efficient exploration [Hu et al., 2022, Terry et al., 2020]. In challenging

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

multi-agent tasks, extensive exploration and diverse policies are crucial [Jiang and Lu, 2021, Li et al., 2021]. For example, in a football game, agents must adopt varied roles and strategies for effective collaboration and goal scoring.

Prior works primarily adopt the mutual information objective to promote diversity among agents [Jiang and Lu, 2021, Li et al., 2021, Jo et al., 2024]. These methods encourage agents to explore diverse individual agent trajectories [Li et al., 2021], observations [Jiang and Lu, 2021], or formations (e.g., observation differences) [Jo et al., 2024]. While these methods do lead to trajectories that are mutually different among agents, the mutual information objective cannot measure how different the induced trajectories are. Slight differences between trajectories are enough to maximize the mutual information objective, which does not necessarily encourage exploration (see Section 3 for more details). To encourage exploration, some recent advances [Hu et al., 2024, Bettini et al., 2024] are based on the Wasserstein distance, a metric-aware measure that quantifies the distance between two distributions. These methods achieve diversity among agents through dynamic parameter sharing [Hu et al., 2024] or policy distribution fusion [Bettini et al., 2024] according to the Wasserstein distance metric. Unfortunately, these methods overlook the similarity of agents' initial policies resulting from shared policy network parameters, which causes the Wasserstein distance, intended to measure policy differences, to converge to zero, leading to ineffective diversity incentives.

To overcome this limitation and enjoy the metric-aware benefit of the Wasserstein distance, we propose a novel Wasserstein Contrastive Diversity (WCD) exploration method, which enlarges the Wasserstein distance in a latent contrastive trajectory representation space. To generate effective diversity incentives based on the Wasserstein distance, we propose a novel next-step prediction method based on Contrastive Predictive Coding (CPC) to learn distinguishable trajectory representations, and calculate the Wasserstein distance between distributions of trajectory representations of different agents.

Our contributions can be summarized as follows: First, because of the similarities among agents' initial policies, the Wasserstein distance cannot provide effective feedback to learn diverse policies. To solve this issue, we consider a trajectory representation space in order to make the Wasserstein distance meaningful. To construct the representation space, we propose a next-step prediction method based on Contrastive Predictive Coding (CPC) [Oord et al., 2018] to learn distinguishable trajectory representations. Second, due to the high computation cost of calculating the Wasserstein distance, we propose a novel Gaussian kernel method to optimize dual functions of the Wasserstein distance, significantly reducing the computational cost. Third, we extend the Wasserstein distance to multiple policy learning by introducing a nearest neighbor intrinsic reward. We further integrate our method with QMIX. Fourth, we show the outperformance of our method against existing state-of-the-art methods by testing it in various challenging multi-agent tasks.

2 Backgrounds

2.1 Multi-Agent System

We consider modeling the fully cooperative multi-agent Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [Oliehoek and Amato, 2015], defined as a tuple $\langle A, S, U, P, R, O, \Omega, \gamma \rangle$. Here, A denotes a set of $|A|$ agents, $s \in S$ represents the global state of the environment, and U stands for the set of agents' 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 $u^a \in U$. All agents' actions collectively form a joint action $\boldsymbol{u}$, leading the environment to transition to the next state s' based on the probability drawn from the transition function $P\left(s' \mid s, \boldsymbol{u}\right)$. Simultaneously, the environment provides the agents with a shared team reward $r = R(s, \boldsymbol{u})$. $\gamma \in [0, 1)$ is the reward discount factor. The observation-action pairs $\langle o^a, u^a \rangle$ of agent a during an episode constitute its trajectory $\tau^a \in \mathcal{T}$. Each agent a learns its individual policy $\pi^a\left(u^a \mid \tau^a\right)$, contributing to the formation of a joint policy $\boldsymbol{\pi}$, aimed at maximizing the joint action-value function $Q^{\boldsymbol{\pi}}(s, \boldsymbol{u}) = \mathbb{E}_{s_{0:\infty}, \boldsymbol{u}_{0:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, \boldsymbol{u}_0 = \boldsymbol{u}, \boldsymbol{\pi}\right]$.

2.2 Wasserstein Distance

The Wasserstein distance formulates an optimal transport problem that measures the distance or discrepancy between two probability distributions [Villani et al., 2009]. Given two probability

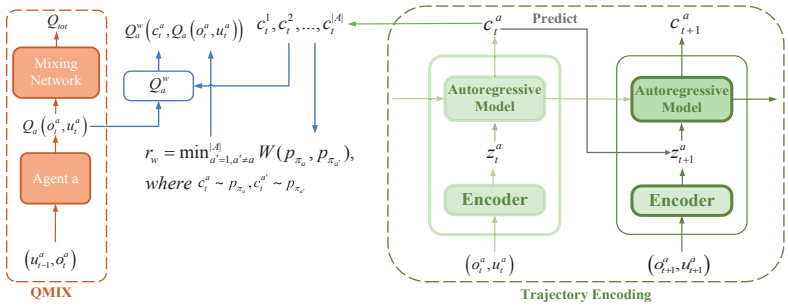


Figure 1: Architecture of WCD. On the left shows the proposed next-step prediction method which learns trajectory representations by predicting the next-step observation-action embedding with Contrastive Predictive Coding (CPC) to make the Wasserstein distance effective. On the right shows the combination of our method with QMIX. We update the policies of agents towards maximizing the intrinsic rewards r_w based on the Wasserstein Distance between the current agent and its nearest neighbor by introducing an additional intrinsic utility network Q_a^w.

distributions p and q over domains $\mathcal{X} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ respectively, the Wasserstein distance with a cost function $c(x, y)$: $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is defined as:

$$\mathcal{W}_c(p, q) = \inf_{\gamma \in \Gamma(p,q)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \mathrm{d}\gamma(x, y) \tag{1}$$

where $\Gamma(p, q)$ is a set of all possible couplings of distributions p and q over the product space $\mathcal{X} \times \mathcal{Y}$. The probability distributions p and q are the marginals of the coupling $\gamma(x, y)$ over space $\mathcal{X}$ and $\mathcal{Y}$, respectively, i.e., $\int_{\mathcal{M}} \gamma(x, y) \mathrm{d}y = p(x)$ and $\int_{\mathcal{M}} \gamma(x, y) \mathrm{d}x = q(y)$.

In practice, we adopt a tractable smoothed Wasserstein distance $\tilde{W}_c(p, q)$ based on Fenchel-Rockafellar duality [Villani et al., 2009],

$$\tilde{W}_c(p, q) = \sup_{\mu, \nu} \mathbb{E}_{x \sim p(x), y \sim q(y)} \left[\mu(x) - \nu(y) - \beta \exp\left(\frac{\mu(x) - \nu(y) - c(x, y)}{\beta} \right) \right] \tag{2}$$

where $\mu : \mathcal{X} \to \mathbb{R}$ and $\nu : \mathcal{Y} \to \mathbb{R}$ are dual functions on continuous domains. β is a smoothing parameter. The dual form of the Wasserstein distance allows for the parametrization of dual functions, thereby mitigating the computational complexity of optimizing the optimal transport problem.

3 Quantitative comparison between the Wasserstein distance and the KL divergence

The mutual information objective is based on the KL divergence, which is metric-agnostic. We next compare the KL divergence and the Wasserstein distance to illustrate that our Wasserstein distance-based method can more efficiently encourage multi-agent diversity compared to the KL divergence. To illustrate the difference between the Wasserstein distance and the KL divergence, we take a Gaussian distribution example. Let $p \sim \mathcal{N}\left(\mu_p, \sigma^2\right)$ and $q \sim \mathcal{N}\left(\mu_q, \sigma^2\right)$. As $\sigma \to 0$, the probability mass of p and q converges to their means, thus we can achieve the KL divergence between two distributions p and q as $\lim_{\sigma \to 0} D_{\mathrm{KL}}(p \| q) = \infty$, which is independent of the specific means μ_p and μ_q. The Wasserstein distance between p and q is $\lim_{\sigma \to 0} W(p, q) = |\mu_p - \mu_q|$. We note that the Wasserstein distance provides an explicit measurement of distance, whereas the KL divergence focuses solely on distinguishability and has no relevance to the metric of the underlying data distribution. Thus, the KL divergence can be easily maximized with slight differences between different agents' trajectories, which does not necessarily encourage visitations of diverse trajectories with large distances. However, due to the metric-aware property of the Wasserstein distance, our method can not only encourage the visitations of different trajectories, as in the KL divergence, but also maximize the distance between diverse trajectories that leads to better trajectory space coverage and more sufficient exploration.

4 Wasserstein Contrastive Diversity

In this section, we detail our proposed Wasserstein Contrastive Diversity (WCD). First, we present how to learn distinguishable trajectory representations to generate effective feedback for the Wasserstein distance. Then, we show how to maximize the Wasserstein distance between different trajectory distributions in the latent representation space.

4.1 Contrastive Predictive Trajectory Representations

In the framework of CTDE, agents that share the same policy network parameters have similar initial policies. Therefore, the Wasserstein distance between any two agents' policy distributions tends to approach zero, i.e., $W(X, Y) \to 0$, where X and Y respectively represent the policy distributions of two agents. Our motivation is to propose a next-step prediction method based on Contrastive Predictive Coding (CPC) [Oord et al., 2018] to learn distinguishable trajectory representations, enabling trajectories induced by different agents' policies to collapse to diverse distributions in a latent representation space. In this space, the Wasserstein distance can effectively encourage the learning of diverse policies.

Initially, as shown in the left of Figure 1, we encode the observation-action pairs $x_t^a = (o_t^a, u_t^a)$ with a non-linear encoder g_{θ_e} into a latent embedding $z_t^a = g_{\theta_e}(x_t^a)$. Then, we use an autoregressive model g_{θ_g} to summarize all the latent embeddings and output the trajectory representation $c_t^a = g_{\theta_g}(z_{\leq t}^a)$ at timestep t. We simply denote $g_\theta = \{g_{\theta_e}, g_{\theta_g}\}$ to represent the overall trajectory encoder. For simplicity, we adopt standard architectures such as MLPs for g_{θ_e} and GRUs for g_{θ_g}.

To train g_θ to learn distinguishable trajectory representations, we model a density ratio that preserves the underlying information between the trajectory representation c_t^a and the next-step observation-action x_{t+1}^a:

$$f\left(x_{t+1}^a, c_t^a\right) \propto \frac{p\left(x_{t+1}^a \mid c_t^a\right)}{p\left(x_{t+1}^a\right)} \tag{3}$$

where $f\left(x_{t+1}^a, c_t^a\right) = \exp(g_{\theta_e}(x_{t+1}^a)^T W c_t^a) = \exp({z_{t+1}^a}^T W c_t^a)$ calculates the similarity between the next-step observation-action embedding z_{t+1}^a and a linear transformation $W^T c_t^a$ with the parameter W used for the next-step prediction. Compared to modeling $p\left(x_{t+1}^a \mid c_t^a\right)$ directly by a generative method that requires to reconstruct every detail in x_{t+1}^a, modeling the density ratio has lower computation cost and is more effective in extracting shared information between x_{t+1}^a and c_t^a. Moreover, we infer the latent embedding z_{t+1}^a instead of the raw x_{t+1}^a, which avoids modeling high-dimensional observation-action space. To let $f\left(x_{t+1}^a, c_t^a\right)$ be proportional to the density ratio, inspired by CPC, given a set of next-step observation-action pairs of all agents $\mathcal{C} = \{x_{t+1}^{a'} = (o_{t+1}^{a'}, u_{t+1}^{a'})\}_{a'=1}^{|A|}$, we minimize a InfoNCE loss [Oord et al., 2018]:

$$\mathcal{L}_N = -\mathop{\mathbb{E}}_{(c_t^a, \mathcal{C})\sim\mathcal{D}}\left[\log \frac{f\left(x_{t+1}^a, c_t^a\right)}{\sum_{x_{t+1}^{a'}\in\mathcal{C}} f\left(x_{t+1}^{a'}, c_t^a\right)}\right] \tag{4}$$

By using the next-step observation-action pairs of other agents as noisy samples in Equation 4 and contrasting the trajectory representation c_t^a with these noises, the trajectory representation c_t^a stays close to its associated next-step observation-action embedding while being far away from other noisy embeddings. As a result, the trajectory encoder g_θ is trained by minimizing the InfoNCE loss to learn distinguishable trajectory representations.

4.2 Wasserstein Distance between Trajectory Representations

We then encourage the exploration of diverse trajectories by maximizing the Wasserstein distance between the trajectory distributions of different agents in a latent representation space. Let p_{π_1} and p_{π_2} be the trajectory representation distributions of agent 1 and agent 2, respectively. The Wasserstein distance between p_{π_1} and p_{π_2} is defined as follows:

$$\tilde{W}_c(p_{\pi_1}, p_{\pi_2}) = \sup_{\mu,\nu} \mathbb{E}_{c_t^1 \sim p_{\pi_1}, c_t^2 \sim p_{\pi_2}} \left[\mu(c_t^1) - \nu(c_t^2) - \beta \exp\left(\frac{\mu(c_t^1) - \nu(c_t^2) - c(c_t^1, c_t^2)}{\beta} \right) \right] \quad (5)$$

where the cost function $c(c_t^1, c_t^2)$ is represented by the Euclidean distance between the points c_t^1 and c_t^2, i.e., $c(c_t^1, c_t^2) = \|c_t^1 - c_t^2\|$. It is notable that to compute the Wasserstein distance, we may simply parameterize dual functions with neural networks like previous works [Pacchiano et al., 2020, Dadashi et al., 2021, He et al., 2022, Park et al., 2024]. However, this may lead to high computational costs in our multi-agent settings, as we need to compute the Wasserstein distance for each pair of agents. To learn optimal dual functions μ and ν to compute the Wasserstein distance with low computational costs, we resort to the kernel method [Hearst et al., 1998] that has been widely used in machine learning. Specifically, we consider representing dual functions with linear combinations of Gaussian kernel functions approximated by the random feature map [Rahimi and Recht, 2007]. For example, let the dual function μ has the following form: $\mu(\mathbf{x}) = (\lambda^\mu)^\top \phi(\mathbf{x})$. For $\mathbf{x} \in \mathbb{R}^d$, $\phi(\mathbf{x}) = \frac{1}{\sqrt{m}} \cos(\mathbf{G}\mathbf{x} + \mathbf{b})$ represents a m-dimensional random feature map, where $\mathbf{G} \in \mathbb{R}^{m \times d}$ is a Gaussian with entries sampled from a normal distribution $\mathcal{N}(0, 1)$ and $\mathbf{b} \in \mathbb{R}^m$ with entries sampled from a uniform distribution $U(0, 2\pi)$. This means that when we optimize the dual function μ, we only need to learn the dual vector $\lambda^\mu \in \mathbb{R}^m$, which significantly reduces the computational cost compared with parameterizing dual functions with computationally intensive neural networks.

To learn optimal dual functions, we perform stochastic gradient descent (SGD) over the Wasserstein distance objective in Equation 5. Given dual functions μ and ν that are modeled by kernels κ and ℓ, respectively, and trajectory representation samples $\{c_t^1, c_t^2\} \sim (p_{\pi_1}, p_{\pi_2})$, we apply the chain rule to Equation 5 and the gradients with respect to λ^μ and λ^ν are

$$\begin{aligned} &\nabla_{(\lambda^\mu, \lambda^\nu)} \tilde{W}_c(p_{\pi_1}, p_{\pi_2}) = \mathbb{E}_{c_t^1 \sim p_{\pi_1}, c_t^2 \sim p_{\pi_2}} \left[(1 - y) \begin{pmatrix} \phi_\kappa(c_t^1) \\ -\phi_\ell(c_t^2) \end{pmatrix} \right], \\ &\text{where } y = \exp\left(\frac{(\lambda^\mu)^\top \phi_\kappa(c_t^1) - (\lambda^\nu)^\top \phi_\ell(c_t^2) - C(c_t^1, c_t^2)}{\beta} \right) \end{aligned} \quad (6)$$

We approximate the expectation by averaging the function values over a batch of trajectory representation samples from the replay buffer that is used to store agent experiences during training.

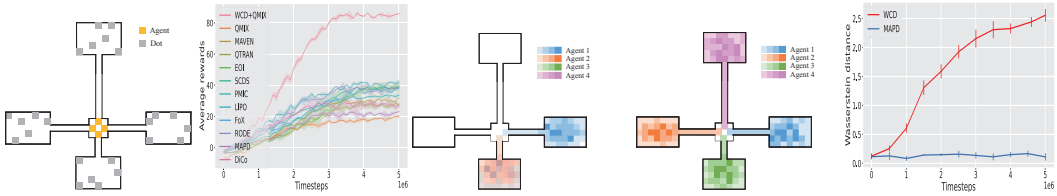
As we have computed the value of the Wasserstein distance, we can view the Wasserstein distance as an intrinsic reward $r_w = W(p_{\pi_1}, p_{\pi_2})$, which enables us to deploy our method in MARL algorithms to maximize the Wasserstein distance. When the number of agents $|A|$ is more than two, the trajectory of an arbitrary agent should keep distance with any other agent. In practice, we empirically find that employing an intrinsic reward $r_w = \min_{a'=1, a' \neq a}^{|A|} W\left(p_{\pi_a}, p_{\pi_{a'}}\right)$ for each agent to keep the trajectory of the current agent a to be away from its nearest neighbor trajectory in a latent representation space can lead to better performance. The pseudocode for our method can be found in Appendix C.

4.3 Practical Learning Algorithm

We next show how to integrate our method with QMIX [Rashid et al., 2018], a state-of-the-art MARL algorithm. QMIX learns optimal individual policies, that maximizes shared team rewards, for agents through optimizing the joint action-value function $Q^{\boldsymbol{\pi}}$ approximated by Q_{tot}, an output of a mixing network that monotonically mixes the agent utilities (where the policies are derived) of all agents. In QMIX, in order to maximize the Wasserstein distance-based intrinsic rewards, we cannot simply add each agent's intrinsic rewards to the shared team reward. More detailed explanations can be found in Appendix A. To integrate our method with QMIX, we additionally introduce an intrinsic utility network Q_a^w, which takes as input the agent utility $Q_a(o_t^a, u_t^a)$ and the trajectory representation c_t^a. We update Q_a^w towards maximizing the intrinsic rewards by minimizing the TD loss as follows

$$\begin{aligned} \mathcal{L}_{TD}^w &= \mathbb{E}_{(o_t^a, u_t^a, o_{t+1}^a) \sim \mathcal{D}} \left[\left(Q_a^w\left(c_t^a, Q_a(o_t^a, u_t^a)\right) - y \right)^2 \right], \\ &where \quad y = r_w + \gamma \bar{Q}_a^w\left(c_{t+1}^a, \bar{Q}_a\left(o_{t+1}^a, u_{t+1}^a\right)\right) \end{aligned} \quad (7)$$

where $\bar{Q}_a^w$ and $\bar{Q}_a$ are target networks employed to stabilize training and $\mathcal{D}$ is the replay buffer for storing trajectory samples. $\mathcal{L}_{TD}^w$ can be seen as a regularizer that introduces an auxiliary gradient



(a) Illustration of Pac-Men (b) Average rewards (c) Visitation heatmap of MAPD (d) Visitation heatmap of WCD (e) Wasserstein distance

Figure 2: Performance comparison between our proposed WCD and baselines in Pac-Men. We report both the mean and standard deviation of the performance tested across five random seeds.

to the agent utility network Q_a in order to learn diverse trajectories. We can thus get the total loss function

$$\mathcal{L}_{total} = \mathcal{L}_{TD} + \alpha \mathcal{L}_{TD}^{w} \tag{8}$$

where $\mathcal{L}_{TD}$ is the TD loss of QMIX to train Q_{tot} and α is a coefficient that changes the weight of $\mathcal{L}_{TD}^{w}$. As $\alpha \to 0$, our method converges to QMIX. Through minimizing $\mathcal{L}_{total}$, we train the overall framework of our method end-to-end in a centralized manner. As a result, agents learn their policies towards maximizing both team rewards and the Wasserstein distance between different agent's trajectory representation distributions. Someone may question whether the regularizer $\mathcal{L}_{TD}^{w}$ conflicts with the global optimum, where agents sometimes need to behave similarly. For example, in SMAC scenarios, agents may be required to fire at the same enemy to quickly defeat it. Empirical results demonstrate that our method does not hinder agents from learning such similar behaviors when they yield greater environmental rewards. This is because our method is essentially an exploration technique, similar to Soft Actor-Critic [Haarnoja et al., 2018] that employs entropy regularization to maximize policy entropy, thereby encouraging sufficient exploration. Numerous studies [Ladosz et al., 2022] have shown that efficient exploration is beneficial for learning optimal policies.

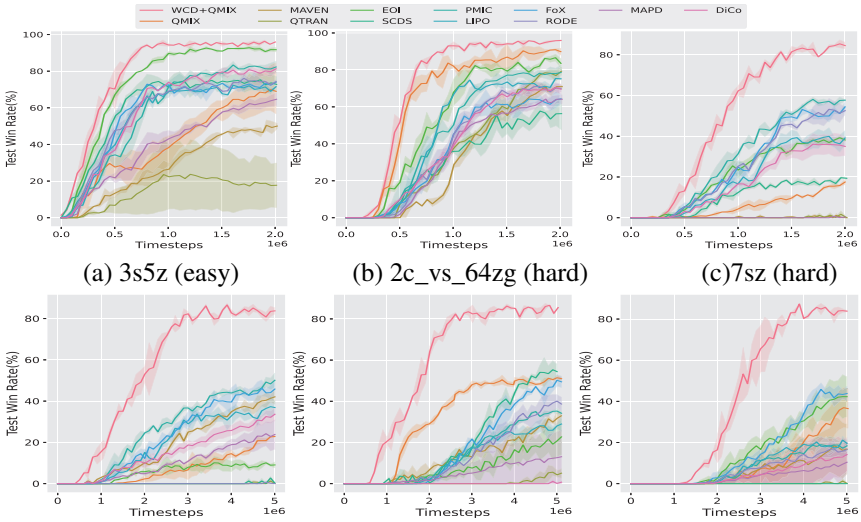
For policy gradient methods, we refer the reader to Appendix D where we integrate our method with the policy gradient-based method MAPPO.

5 Experiments

In this section, we use challenging multi-agent tasks from Pac-Men, SMAC, and SMACv2 to demonstrate the outperformance of our method. We show comparison of our method against the state-of-the-art methods such as value-decomposition methods (QMIX [Rashid et al., 2018] and QTRAN [Son et al., 2019]), role-based diversity methods (RODE [Wang et al., 2020c]), mutual information-based diversity methods (MAVEN [Mahajan et al., 2019], EOI [Jiang and Lu, 2021], SCDS [Li et al., 2021], PMIC [Li et al., 2022], LIPO [Charakorn et al., 2023], and FoX [Jo et al., 2024]), and Wasserstein distance-based diversity methods (MAPD [Hu et al., 2024] and DiCo [Bettini et al., 2024]). Without loss of generality, the comparison results are shown with both the mean and standard deviation of the performance tested across five random seeds. For a fair comparison, we adopt the same hyperparameters and policy network architecture across all methods. More training details and hyperparameters are provided in Appendix K.

5.1 Pac-Men

We first test our method in Pac-Men, as illustrated in Figure 2a, to investigate the effectiveness of our method in encouraging multi-agent diversity. Pac-Men is a foraging game, where four agents initialized at the center of the maze try to eat the dots randomly distributed in four edge rooms. Agents can move to these rooms along paths of different lengths. Each agent only has a partial observation of 4×4 grid around them. The goal of the agent is to collect as many dots as possible to achieve more rewards. Notably, agents arriving at the same edge room may result in inefficient competition. They are expected to behave differently and move to different rooms.



(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)

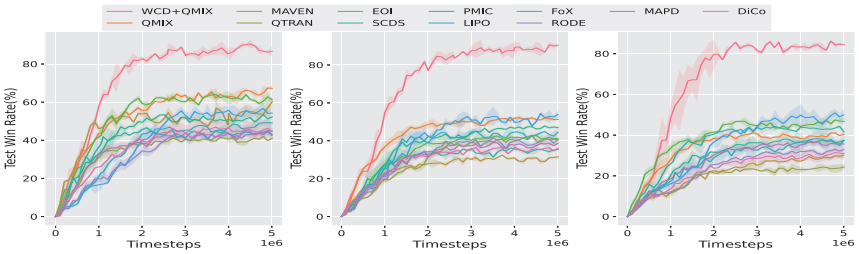
Figure 3: Performance comparison between our proposed WCD and baselines in the SMAC scenarios.

The results shown in Figure 2b demonstrate the outperformance of our method compared to baselines. Through maximizing the Wasserstein distance between different trajectory distributions in a latent space, agents respectively move to the four edge rooms, as depicted by Figure 2d, leading to diverse policies and efficient cooperation. MAPD fails to learn diverse policies. As shown in Figure 2c, some agents adopt the same policy and move to the same edge room, resulting in poor performance. From Figure 2e, we note that MAPD does not provide effective diversity measures in terms of the average Wasserstein distance, which does not necessarily encourage exploration of diverse policies. However, by learning a contrastive trajectory representation space through the proposed next-step prediction method, our method produces effective Wasserstein distance as intrinsic rewards to learn diverse policies. Some mutual information-based baselines such as EOI and SCDS, employing the metric-agnostic variational intrinsic reward, achieve similar performance. They may not find the edge room with the longest path, leading to sub-optimal performance. This is because the variational intrinsic reward converges quickly due to its metric-agnostic property, leading to insufficient incentives for exploration. Conversely, our Wasserstein distance-based metric-aware intrinsic reward can continuously provide effective reward signals for agents to encourage sufficient exploration.

5.2 SMAC

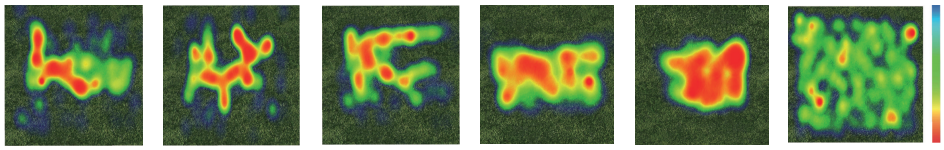
We then test our method on the StarCraft Multi-Agent Challenge (SMAC) [Samvelyan et al., 2019], a commonly used benchmark for evaluating cooperative MARL algorithms, consisting of various combat scenarios with different difficulties. We evaluate our method in 6 scenarios of SMAC including 3s5z (easy), 2c_vs_64zg (hard), 7sz (hard), 6h_vs_8z (super hard), corridor (super hard), and 3s5z_vs_3s6z (super hard). The version of SMAC adopted in our experiments is SC2.4.10. The performance comparison are not applicable across different SMAC versions.

As shown in Figure 3, our method maintains its outperformance in both easy and hard scenarios and significantly outperforms all baselines in the super hard scenarios. QMIX struggles to learn optimal cooperative policies in the super hard scenarios. However, our method can efficiently improve the performance of QMIX by encouraging multi-agent diversity. Moreover, our method shows a substantial performance boost compared to MAPD, highlighting the effectiveness of the Wasserstein distance in the contrastive trajectory representation space. MAPD even achieves worse performance than the mutual information-based methods in the super hard scenarios. This phenomenon is due to the fact that the Wasserstein distance metric employed in MAPD may not work properly under the policy network parameter-sharing setting. Compared to mutual information-based methods, our method achieves better performance due to the maximization of the metric-aware Wasserstein distance in the contrastive trajectory representation space, leading to more sufficient exploration. We further present visualization examples of diverse policies learned by our method in the super hard



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

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



(a) MAVEN (b) EOI (c)SCDS (d) LIPO (e) MAPD (f) WCD

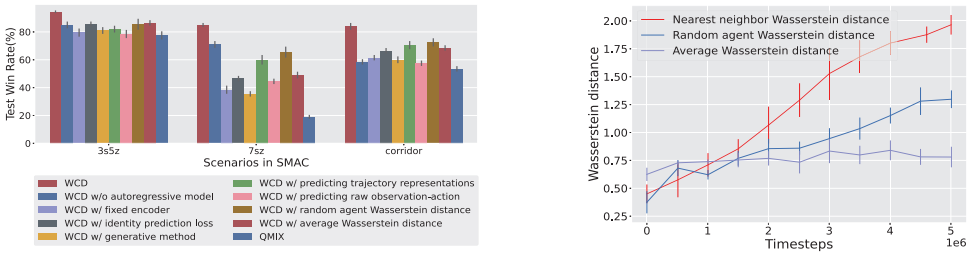
Figure 5: Visitation heatmaps of different algorithms in the terran_5_vs_5 scenario.

scenarios in Appendix L. The mutual information-based methods may not enable agents to learn trajectories with large variations. EOI does not result in satisfactory performance as the trajectory classifier employed in EOI overfits the agent identity information, impeding further exploration. Moreover, it is notable that our method also achieves satisfactory performance in the easy 3s5z scenario where agents sometimes need to behave in the same way to master the trick of 'focus fire', demonstrating that our method would not prevent the homogeneous behaviors that can lead to more environmental rewards. More experimental results related to such homogeneous behaviors can be found in Appendix E.2. These results reveal that our method efficiently balances exploration and exploitation, resulting in the learning of optimal cooperative policies.

Stochasticity and Exploration Although SMAC consists of many challenging scenarios, the agents may overfit the timesteps regardless of real environmental states Ellis et al. [2022] since the team compositions and the initial positions of units are the same in each episode. We further adopt the SMACv2 benchmark Ellis et al. [2022]. SMACv2 introduces stochasticity by deploying random team compositions and random initial positions, which challenges agents to continuously explore optimal policies. The performance comparison of our method against baselines are shown in Figure 4. Our method achieves superior performance in all scenarios compared to the baselines. Our method significantly improves the performance of QMIX by introducing effective Wasserstein distance objective as a regularizer to encourage multi-agent diversity. The mutual information-based methods do not yield satisfactory performance. We believe this is because the variational intrinsic reward adopted in these methods converge quickly when the trajectories of different agents are identified. As a result, it cannot provide effective feedback for agents to continuously explore. Moreover, MAPD fails to yield sufficient exploration to adapt to the environment stochasticity due to the ineffective Wasserstein distance incentives. Instead, our method can continuously provide efficient Wasserstein distance-based intrinsic rewards to encourage exploration. This can be verified by the visitation heatmaps of agents trained by various methods shown in Figure 5. We observe that agents trained by our method achieve more extensive environmental exploration compared to those trained using baselines distributed only in partial areas.

5.3 Ablation Study

We conduct ablation studies to evaluate the contributions of the main components in our method. To test the contribution of the autoregressive model employed to learn trajectory representations, we ablate the autoregressive model and only use the non-linear encoder g_{θ_e} regardless of the trajectory context. To measure the contribution of the representation learning method, we design five variants: (i) employing a randomly initialized encoder with fixed parameters for encoding trajectories, (ii) learning trajectory representations by directly predicting the agent identities of various trajectories



(a) (b)

Figure 6: (a) Performance comparison of different variants in the scenarios of SMAC. (b) Different kinds of Wasserstein distances.

instead of employing the InfoNCE loss, (iii) learning trajectory representations by adopting a generative method to model $p\left(x_{t+1}^a \mid c_t^a\right)$ instead of modeling the density ratio, (iv) predicting the trajectory representation c_{t+1}^a instead of the latent embedding z_{t+1}^a, and (v) directly predicting the raw observation-action x_{t+1}^a. To test the Wasserstein distance objective, we ablate the nearest neighbor intrinsic reward r_w and use the Wasserstein distance between trajectory representation distributions of the current agent and another randomly selected agent, and the average Wasserstein distance as intrinsic rewards, respectively.

We test these variants in the scenarios from SMAC, and the results are shown in Figure 6a. We note that the absence of any of the components employed in our method results in significant performance degradation. Encoding trajectory representations with a fixed encoder leads to poor performance, demonstrating the importance of learning distinguishable trajectory representations. Moreover, learning trajectory representations by minimizing the identity prediction loss or learning a generative model is less efficient than our method. These methods do not necessarily learn distinguishable trajectory representations with large variations, thus the representations may not work properly in the Wasserstein distance objective to produce efficient feedback. Also, using the generative method leads to lower learning efficiency due to high computational cost. Predicting the trajectory representations or the raw observation-action does not lead to better performance than predicting the latent embeddings adopted in our method. The average Wasserstein distance does not yield satisfactory performance and even achieves worse performance than the random agent Wasserstein distance. As shown in Figure 6b, the average Wasserstein distance intrinsic rewards do not provide effective incentives to encourage multi-agent diversity. Instead, our nearest neighbor Wasserstein distance is more sensitive to the trajectory representation variations. Despite the performance degradation caused by different kinds of Wasserstein distances, these Wasserstein distance methods also lead to significant performance improvement over QMIX, demonstrating the robustness of our representation learning method. As the difficulty of the task increases, we note obvious performance degradation caused by the ablation of the autoregressive model, indicating that learning trajectory representations results in more robust performance, especially in hard tasks.

6 Related Works

Diversity within MARL aims to learn diversified policies among agents to encourage efficient exploration. To achieve this goal, numerous diversity-driven methods have proposed different intrinsic motivations or regularizers. RODE [Wang et al., 2020c] promotes diversity by assigning distinct actions to predefined roles; however, its effectiveness may decrease in scenarios with continuous actions and extensive action spaces. MAVEN [Mahajan et al., 2019] introduces a value-based approach that conditions agents' joint behaviors on a shared latent variable controlled by a hierarchical policy. EOI [Jiang and Lu, 2021] utilizes a supervised learning approach to promote agent individuality, employing a probabilistic classifier to predict agents' probability distributions based on their observations. SCDS [Li et al., 2021] concentrates on enhancing multi-agent diversity by optimizing mutual information between agent identities and trajectories. PMIC [Li et al., 2022] adopts a unique approach by maximizing the mutual information concerning superior cooperative behaviors while minimizing it regarding inferior behaviors. LIPO [Charakorn et al., 2023] uses policy compatibility as a proxy to learn diverse policies and diversifies agents' behaviors through the mutual information

objective. FoX [Jo et al., 2024] proposes formation-based exploration, encouraging visitations of diverse formations by guiding agents to fully understand their current formations. Although these approaches show promise in enhancing multi-agent diversity, the KL divergence derived from the mutual information objective may lead to insufficient exploration.

Wasserstein distance, emerging as an advanced measure of distribution dissimilarity, has garnered attentions of researchers from the community of machine learning. Many generative models [Arjovsky et al., 2017, Ambrogioni et al., 2018, Patrini et al., 2020, Tolstikhin et al., 2018] have incorporated the Wasserstein distance objective and demonstrate the effectiveness of Wasserstein distance in scenarios where distributions become degenerate on a sub-manifold within pixel space. Some recent advances [Hu et al., 2024, Bettini et al., 2024] rely on the maximization of the Wasserstein distance to enlarge the policy differences among agents. MAPD [Hu et al., 2024] introduces the use of the Wasserstein distance as a metric for measuring policy differences. It normalizes the action distributions of different agents and computes the Wasserstein distance between them. Different from MAPD, our method learns a contrastive trajectory representation space by contrasting the trajectories of different agents to learn distinguishable trajectory representations to make the Wasserstein distance meaningful. Empirical results shown in Section 5 demonstrates the superiority of our proposed representation learning method compared to MAPD. DiCo [Bettini et al., 2024] proposes using a metric based on the average Wasserstein distance to control the diversity among agents. It performs well in a simple multi-agent navigation task. However, it may suffer from insufficient exploration in challenging multi-agent tasks due to the controlled diversity. Furthermore, these methods fail to capture the similarities between agent policies, resulting in an ineffective use of the Wasserstein distance, which ultimately impacts the performance of the proposed methods.

7 Limitations and Future Directions

The cost function of the Wasserstein distance determines how the probability mass is transferred. For simplicity, we resort to the Euclidean distance as the cost function of the Wasserstein distance in the experimental settings. However, selecting an optimal cost function for the Wasserstein distance to address specific multi-agent tasks still remains a challenge.

8 Conclusion

In this paper, we propose a novel WCD exploration method. Unlike previous Wasserstein distance-based methods, our method maximizes the Wasserstein distance between the trajectory distributions of different agents in a contrastive trajectory representation space learned by a next-step prediction method, leading to sufficient exploration. We deploy our method in MARL by introducing a nearest neighbor intrinsic reward based on the Wasserstein distance. The experimental results demonstrate that our method learns more diverse policies and leads to more sufficient exploration compared to the baseline methods. This simple yet effective method provides a novel idea of learning useful representations to promote exploration, which shows promising results in learning cooperative policies for challenging multi-agent tasks.

Acknowledgments and Disclosure of Funding

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).

References

L. Ambrogioni, U. Güçlü, Y. Güçlütürk, M. Hinne, M. A. van Gerven, and E. Maris. Wasserstein variational inference. *Advances in Neural Information Processing Systems*, 31, 2018.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

M. Bettini, R. Kortvelesy, and A. Prorok. Controlling behavioral diversity in multi-agent reinforcement learning. *arXiv preprint arXiv:2405.15054*, 2024.

Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2012.

R. Charakorn, P. Manoonpong, and N. Dilokthanakul. Generating diverse cooperative agents by learning incompatible policies. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=UkU05GOH7_6`.

R. Dadashi, L. Hussenot, M. Geist, and O. Pietquin. Primal wasserstein imitation learning. In *ICLR 2021-Ninth International Conference on Learning Representations*, 2021.

B. Ellis, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. N. Foerster, and S. Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2212.07489*, 2022.

A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

S. He, Y. Jiang, H. Zhang, J. Shao, and X. Ji. Wasserstein unsupervised reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6884–6892, 2022.

M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

S. Hu, C. Xie, X. Liang, and X. Chang. Policy diagnosis via measuring role diversity in cooperative multi-agent rl. In *International Conference on Machine Learning*, pages 9041–9071. PMLR, 2022.

T. Hu, Z. Pu, X. Ai, T. Qiu, and J. Yi. Measuring policy distance for multi-agent reinforcement learning. *arXiv preprint arXiv:2401.11257*, 2024.

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

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

P. Ladosz, L. Weng, M. Kim, and H. Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

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

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

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

X. Ma, Y. Yang, C. Li, Y. Lu, Q. Zhao, and J. Yang. Modeling the interaction between agents in cooperative multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 853–861, 2021.

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

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

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

A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

A. Pacchiano, J. Parker-Holder, Y. Tang, K. Choromanski, A. Choromanska, and M. Jordan. Learning to score behaviors for guided policy optimization. In *International Conference on Machine Learning*, pages 7445–7454. PMLR, 2020.

S. Park, O. Rybkin, and S. Levine. METRA: Scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=c5pwL0Soay`.

G. Patrini, R. Van den Berg, P. Forre, M. Carioni, S. Bhargav, M. Welling, T. Genewein, and F. Nielsen. Sinkhorn autoencoders. In *Uncertainty in Artificial Intelligence*, pages 733–743. PMLR, 2020.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

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

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

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

P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al. 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*, pages 2085–2087, 2018.

J. K. Terry, N. Grammel, S. Son, B. Black, and A. Agrawal. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625*, 2020.

I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. In *6th International Conference on Learning Representations (ICLR 2018)*. OpenReview. net, 2018.

C. Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

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

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

T. Wang, H. Dong, V. Lesser, and C. Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020b.

T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson, and C. Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020c.

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

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

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and a complete (and correct) proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the training details and hyperparameter settings needed to reproduce the experimental results in Appendix K.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the source code in the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the training details in Appendix K.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report both the mean and standard deviation of performance results, averaged over five random seeds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computation platform used in our work in Appendix K.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work aims at advancing the field of Machine Learning and has no societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited the benchmarks used in our work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes] ,

    Justification: We provide the source code in the supplemental material with the documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A The TD loss of QMIX

The TD loss of QMIX to learn the optimal Q_{tot} is defined as:

$$\mathcal{L}_{TD} = \sum_{i=1}^{b} \left[\left(r + \gamma \max_{\mathbf{u}_{t+1}} \bar{Q}_{tot} \left(s_{t+1}, \mathbf{u}_{t+1} \right) - Q_{tot}(s_t, \mathbf{u}_t) \right)^2 \right] \tag{9}$$

where $\bar{Q}_{tot}$ is the target network and b is the size of transition samples from the replay buffer $\mathcal{D}$. r is the global reward shared among agents. Note that since all agent's policies are jointly trained by minimizing the TD loss, we cannot simply apply each agent's intrinsic reward r_w to the global reward r to formulate a reward-shaping to independently train each agent's individual policy. That is why we need to learn an additional intrinsic utility network to maximize the intrinsic reward r_w.

B Theoretical analysis of convergence guarantee of our method

We next provide a theoretical analysis to guarantee the convergence of the MARL when our Wasserstein distance-based intrinsic reward is introduced. To achieve this, we theoretically analysis the boundeness of the intrinsic rewards, which is a standard and critical condition for proving convergence. The bounded rewards state that the rewards received by the agents at each time step are limited within a finite range. For our intrinsic reward, boundedness ensures that the diversity-seeking objective does not overwhelm the primary goal of maximizing the team reward. If the intrinsic reward were unbounded, it could lead to policies that are arbitrarily diverse but suboptimal in terms of task performance. Specifically, for our proposed WCD method, we prove that the Wasserstein distance-based intrinsic reward is bounded. The boundedness of this intrinsic reward is guaranteed by the properties of the learned representation space and the Wasserstein distance itself.

1. Boundedness of the Compact Representation Space

The trajectory representations are guaranteed to be in a compact (i.e., closed and bounded) set due to the architecture of the trajectory encoder. The encoder is composed of a multi-layer perceptron (MLP) and a Gated Recurrent Unit (GRU), both of which use activation functions that constrain their outputs.

Let the trajectory encoder be represented by the function $g_\theta : \mathcal{X} \to \mathcal{C}$, where $\mathcal{X}$ is the space of trajectory histories and $\mathcal{C}$ is the latent representation space. The encoder is a composition of the non-linear encoder g_{θ_e} and the autoregressive model g_{θ_g}.

$$c_t^a = g_{\theta_g}(g_{\theta_e}(x_0^a), ..., g_{\theta_e}(x_{t-1}^a)) \tag{10}$$

The boundedness of the output c_t^a can be shown by analyzing the components:

a. Bounded Activation Functions: The MLP and GRU components of the encoder use activation functions such as the sigmoid ($\sigma(x) = \frac{1}{1+e^{-x}}$) or the hyperbolic tangent ($\tanh(x)$), which are inherently bounded. For any input x, their outputs are confined to a specific range (e.g., $[0, 1]$ for sigmoid, $[-1, 1]$ for tanh).

b. Bounded Weights and Biases: During training, the weights and biases of the neural networks are typically regularized (e.g., through weight decay), which keeps them within a finite range.

Given that the inputs to each layer are transformations of the outputs from previous layers with bounded activation functions, the final output of the network, which is the trajectory representation c_t^a, will also be bounded. If we assume the maximum possible value for any dimension of the representation vector is C_{max}, then the representation space is a subset of a hypercube in $\mathbb{R}^d$:

$$\mathcal{C} \subset [-C_{max}, C_{max}]^d \tag{11}$$

where d is the dimension of the representation space. Since this set is closed and bounded, the representation space is compact.

2. Boundedness of the Wasserstein Distance

The Wasserstein distance is used to measure the cost of optimal transport between two probability distributions. When these distributions are defined over a compact set, the Wasserstein distance is also bounded.

The 1-Wasserstein distance has a dual representation known as the Kantorovich-Rubinstein duality, which is particularly useful for understanding its properties:

$$W_1(p, q) = \sup_{\|f\|_L \leq 1} |\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)]| \tag{12}$$

where the supremum is taken over all 1-Lipschitz functions f.

Let p and q be two probability distributions over the compact representation space $\mathcal{C}$. Since $\mathcal{C}$ is bounded, there exists a constant M such that for any two points $c_1, c_2 \in \mathcal{C}$, the distance between them is bounded: $d(c_1, c_2) \leq M$.

For any 1-Lipschitz function f and any two points $c_1, c_2 \in \mathcal{C}$, we have:

$$|f(c_1) - f(c_2)| \leq d(c_1, c_2) \leq M \tag{13}$$

Now, consider the expectation of f under the distributions p and q.

$$|\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)]| = \left| \int_{\mathcal{C}} f(x)dp(x) - \int_{\mathcal{C}} f(y)dq(y) \right| \tag{14}$$

By choosing a reference point $c_0 \in \mathcal{C}$, we can write:

$$\left| \int (f(x) - f(c_0))dp(x) - \int (f(y) - f(c_0))dq(y) \right| \tag{15}$$

Since $|f(c) - f(c_0)| \leq d(c, c_0) \leq M$ for any $c \in \mathcal{C}$, we have:

$$|\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)]| \leq \int M dp(x) + \int M dq(y) = 2M \tag{16}$$

This shows that the Wasserstein distance is bounded by the diameter of the compact set on which the distributions are supported. The intrinsic reward r_w, being the minimum of these bounded distances, is therefore also bounded. This formalizes the boundedness assumption, which is essential for the stability and convergence of the WCD algorithm.

C Pseudocode for WCD

The pseudocode for WCD is given in Algorithm 1.

D Integrating WCD with the policy-based method

We have implemented our method with the value-based method QMIX. Here, we illustrate the integration of our proposed WCD with policy-based methods. Specifically, we integrate WCD with MAPPO, a state-of-the-art policy-based MARL algorithm measured by SMAC. In MAPPO, all agents share an actor network and a critic network. As each agent learns its own critic, we can straightforwardly incorporate a shaped reward $r_{env} + \alpha r_w$ (where r_{env} represents the environmental reward and r_w denotes the Wasserstein distance-based intrinsic reward) when computing the reward-to-go $\hat{R}$ for updating each agent's critic network. The remaining components of MAPPO do not require modification. We conduct experiments on Pac-Men, SMAC, and SMACv2 to test the performance of WCD+MAPPO. The results, presented in Table 1, demonstrate the superior performance of WCD+MAPPO compared to the baselines.

Algorithm 1 Wasserstein Contrastive Diversity (WCD)

Initialize dual functions μ and ν. Initialize the joint policy $\boldsymbol{\pi} = \{\pi_a\}_{a=1}^{|A|}$.
Randomly initialize Q_{tot} for QMIX.
repeat
  for $each\ episode$ **do**
    Collect the trajectories of all agents $\boldsymbol{\tau}$ induced by the joint policy $\boldsymbol{\pi}$.
    Store them into a replay buffer D.
  end for
  Sample a batch of trajectories $\boldsymbol{\tau}$ from the replay buffer D.
  Train the trajectory encoder g_θ to learn trajectory representations by minimizing the InfoNCE loss given by Equation 4.
  Train dual functions μ and ν by SGD with the gradient given by Equation 6.
  Compute the intrinsic reward $r_w = \min_{a'=1,a'\neq a}^{|A|} W\left(p_{\pi_a}, p_{\pi_{a'}}\right)$ for each agent.
  Jointly train the policy π_a for each agent by minimizing $\mathcal{L}_{total} = \mathcal{L}_{TD} + \alpha\mathcal{L}_{TD}^{w}$.
until Q_{tot} is $converged$

E Environmental details and Additional experimental results

E.1 Environmental details and experimental results

In Pac-Men, four agents are initialized in the central room of a maze. Each agent is restricted to observing a 4×4 grid surrounding them. Randomly distributed dots are present in each edge room. The objective for the agent is to gather as many dots as possible in each edge room. We vary the lengths of paths to evaluate the exploration of environments. Specifically, path lengths for downward, leftward, rightward, and upward directions are set to 3, 6, 6, and 10, respectively. Only one path falls within the agent's observation scope. Dots in each room will respawn once all have been consumed by agents. Agents receive an environmental reward equal to the total number of dots consumed in each time step.

The SMAC benchmark includes many cooperative tasks based on Blizzard's real-time strategy game StarCraft II, designed to evaluate the efficacy of different Multi-Agent Reinforcement Learning (MARL) algorithms. Agent-level control in SMAC utilizes the Machine Learning APIs provided by StarCraft II and DeepMind's PySC2. Each task presents a combat scenario with two armies: one led by allied RL agents and the other by a non-learning game AI. The game ends when all units from any army perish or a predefined time limit is reached. The objective for allied agents is to maximize the game's win rate. To achieve this, agents must learn a sequence of actions to effectively collaborate with allies in vanquishing enemy forces. An illustrative example of such collaboration involves mastering kiting skills, where agents organize formations based on armor types to lure enemy units into pursuit while maintaining a safe distance to minimize damage. The SC2.4.10 version of StarCraft II is utilized, and performance comparison across different versions are not applicable. Experiments are conducted across six scenarios, including 3s5z, 2c_vs_64zg, 7sz, 6h_vs_8z, corridor, and 3s5z_vs_3s6z, spanning various difficulty levels.

SMAC is greatly limited by its lack of stochasticity. To remedy this, the newly released SMACv2 proposes modifications such as incorporating random team compositions and random start positions. These adjustments aim to inject more stochastic elements into the environment to effectively evaluate the exploration capabilities of MARL algorithms. We conduct experiments in three SMACv2 scenarios: terran_5_vs_5, protoss_5_vs_5, and zerg_5_vs_5. In SMACv2, each race in the game of StarCraft II employs three unit types, with units algorithmically assembled into teams. The probability of each unit type appearing in each episode remains fixed throughout training and testing phases. Allied agents have the same unit types as their adversaries. In each episode, allied agents are randomly deployed on the map using either a reflect or surround style.

We present the average returns of all algorithms in Pac-Men, SMAC, and SMACv2, along with their standard deviation over five random seeds, in Table 1. The results indicate the significant performance superiority of our method over baseline methods.

Table 1: Average returns of all algorithms in Pac-Men, SMAC, and SMACv2. ± denotes the standard deviation over five random seeds.

Method	Pac-Men	SMAC						SMACv2		
		3s5z	2c_vs_64zg	7sz	6h_vs_8z	corridor	3s5z_vs_3s6z	terran_5_vs_5	protoss_5_vs_5	zerg_5_vs_5
QMIX	0.21±0.04	0.72±0.13	0.85±0.08	0.17±0.02	0.23±0.03	0.57±0.07	0.36±0.12	0.68±0.03	0.53±0.05	0.41±0.04
MAPPO	0.49±0.03	0.81±0.05	0.83±0.04	0.52±0.06	0.53±0.03	0.62±0.05	0.57±0.08	0.52±0.04	0.47±0.03	0.37±0.03
MAVEN	0.32±0.06	0.51±0.21	0.72±0.06	0.00±0.00	0.42±0.04	0.36±0.08	0.18±0.15	0.58±0.04	0.31±0.05	0.29±0.03
EOI	0.41±0.05	0.87±0.07	0.83±0.02	0.37±0.03	0.08±0.03	0.25±0.11	0.42±0.13	0.65±0.05	0.42±0.03	0.47±0.04
QTRAN	0.28±0.08	0.21±0.19	0.75±0.05	0.00±0.00	0.02±0.02	0.08±0.07	0.02±0.01	0.42±0.02	0.40±0.04	0.25±0.02
SCDS	0.37±0.05	0.76±0.07	0.57±0.09	0.21±0.03	0.03±0.01	0.56±0.06	0.00±0.00	0.52±0.03	0.47±0.05	0.38±0.04
PMIC	0.34±0.03	0.82±0.03	0.79±0.05	0.58±0.02	0.51±0.05	0.37±0.03	0.18±0.06	0.47±0.03	0.36±0.02	0.42±0.02
LIPO	0.43±0.02	0.71±0.03	0.76±0.02	0.39±0.04	0.36±0.06	0.27±0.03	0.21±0.03	0.43±0.02	0.46±0.03	0.37±0.03
FoX	0.39±0.03	0.74±0.02	0.64±0.05	0.56±0.03	0.45±0.05	0.52±0.04	0.43±0.04	0.54±0.03	0.56±0.02	0.49±0.02
RODE	0.37±0.02	0.72±0.03	0.69±0.02	0.54±0.03	0.03±0.02	0.40±0.05	0.17±0.03	0.43±0.02	0.40±0.03	0.34±0.03
MAPD	0.23±0.03	0.63±0.04	0.65±0.03	0.04±0.02	0.26±0.07	0.12±0.08	0.10±0.06	0.44±0.03	0.38±0.02	0.36±0.03
DiCo	0.27±0.02	0.82±0.03	0.71±0.03	0.56±0.03	0.34±0.04	0.05±0.04	0.14±0.08	0.45±0.04	0.36±0.02	0.31±0.02
WCD+QMIX	**0.87±0.03**	**0.95±0.03**	**0.96±0.02**	0.87±0.04	**0.83±0.03**	0.85±0.04	0.82±0.03	0.85±0.03	**0.90±0.02**	0.84±0.03
WCD+MAPPO	0.82±0.02	0.93±0.02	0.89±0.05	**0.94±0.03**	0.79±0.04	**0.87±0.05**	**0.89±0.04**	**0.89±0.03**	0.82±0.02	**0.91±0.04**

E.2 Additional results

Google Research Football We also evaluate our method in three scenarios from Google Research Football (GRF), a challenging, physics-based environment that simulates a football game. In this setting, agents must learn strategic planning, coordination, and precise timing to succeed. The players on the left side (excluding the goalkeeper) serve as agents, trained to develop cooperative policies, while the right-side players are controlled by the game engine. The agents operate within a discrete action space of 19 options, which includes moving in eight directions, sliding, shooting, and passing. Observations for each agent consist of the positions and movement directions of the agent itself, other agents, and the ball. The results of the comparison are presented in Table 2, where our method consistently outperforms the baseline methods across all scenarios.

Table 2: Performance comparisons of our method against the baseline methods in Google Research Football.

Method	academy_3_vs_1_with_keeper	academy_4_vs_2_with_keeper	academy_counter_attack_hard
QMIX	0.23±0.05	0.13±0.09	0.17±0.03
MAPPO	0.31±0.09	0.18±0.09	0.23±0.07
MAVEN	0.18±0.06	0.08±0.06	0.13±0.09
EOI	0.17±0.05	0.05±0.03	0.07±0.03
QTRAN	0.25±0.03	0.13±0.08	0.11±0.05
SCDS	0.42±0.13	0.25±0.11	0.47±0.06
PMIC	0.23±0.08	0.11±0.07	0.16±0.07
LIPO	0.19±0.05	0.07±0.03	0.12±0.05
FoX	0.57±0.05	0.41±0.13	0.33±0.08
RODE	0.37± 0.08	0.16±0.10	0.28±0.06
MAPD	0.23±0.11	0.11±0.06	0.19±0.07
DiCo	0.42 ±0.06	0.29±0.17	0.21±0.12
WCD+QMIX	**0.82±0.11**	0.68±0.13	**0.76±0.09**
WCD+MAPPO	0.78±0.15	**0.75±0.09**	0.63±0.11

Homogeneous behaviors Agents may sometimes desire to behave in the same way. For instance, allied agents in the scenarios of SMAC might take the same action to fire at the same enemy in order to rapidly defeat it. In this section, to demonstrate the effectiveness of our method in learning such behaviors, we evaluate our method in four homogeneous scenarios of SMAC that require the trick of focus fire. The results are shown in Table 3. Our method outperforms QMIX across all scenarios, demonstrating that our method would not prevent the homogeneous behaviors if they can lead to more environmental rewards. In contrast, our method encourages sufficient exploration to search for such optimal cooperative behaviors.

Scalability The scalability of the MARL algorithms refers to their ability to effectively handle the growing number of agents in the environment. The action space grows exponentially with the number of agents, highlighting the urgent need for exploration. In this section, we evaluate the scalability of our method in four scenarios of SMACv2 with an increasing number of agents: terran_5_vs_5,

Table 3: Performance of our method and QMIX in homogeneous scenarios.

Method	8m	5m_vs_6m	8m_vs_9m	10m_vs_11m
WCD+QMIX	0.94±0.02	0.95±0.03	0.93±0.04	0.91± 0.03
QMIX	0.87±0.03	0.65±0.04	0.58±0.05	0.43±0.04

terran_10_vs_10, terran_15_vs_15, and terran_20_vs_20. We present the results in Table 4. Our method maintains its outperformance over QMIX across all scenarios. QMIX suffers from poor scalability due to limited exploration, while our method scales well with an increasing number of agents, demonstrating that our method can lead to sufficient exploration of action space by enlarging the Wasserstein distance between trajectory distributions of different agents in the latent representation space.

Table 4: 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
WCD+QMIX	0.85±0.03	0.86 ±0.02	0.83 ±0.04	0.81 ±0.05
QMIX	0.68±0.03	0.39±0.04	0.24 ±0.06	0.11±0.05

F Comparison with ϵ-Greedy

The ϵ-greedy method is a commonly used exploration strategy in many RL algorithms. Typically, increasing the value of ϵ enhances exploration. In this section, we compare our Wasserstein distance-based method with ϵ-greedy to highlight its effectiveness in promoting exploration within MARL. For this comparison, we set the ϵ values to 0.05, 0.075, and 0.1 for QMIX, and evaluate these settings in the challenging scenarios including corridor, 3s5z_vs_3s6z, terran_5_vs_5, and protoss_5_vs_5. The results, presented in Table 5, show that our Wasserstein distance-based method is more effective in fostering exploration compared to simply increasing ϵ. Notably, increasing ϵ values does not lead to significant performance gains. In multi-agent settings, higher ϵ values primarily increase randomness in an individual agent's action selection without enhancing diversity or coordination among agents, as they fail to consider the trajectories of other agents, resulting in inefficient exploration.

Table 5: Comparison of performance between our method and QMIX using various ϵ values

Method	corridor	3s5z_vs_3s6z	terran_5_vs_5	protoss_5_vs_5
ϵ = 0.05 (QMIX)	0.57 ±0.07	0.36 ±0.12	0.68 ±0.03	0.53 ±0.05
ϵ = 0.075 (QMIX)	0.61 ±0.04	0.39 ±0.11	0.72 ±0.04	0.62 ±0.07
ϵ = 0.1 (QMIX)	0.63 ±0.06	0.44 ± 0.15	0.74 ±0.03	0.69 ±0.06
Wasserstein distance (our method)	0.85 ± 0.04	0.82 ± 0.03	0.85 ±0.03	0.90 ±0.02

G Evaluations of different kernel functions

We use the Gaussian kernel by default in our paper. We may also use a linear kernel to parameterize dual functions. To evaluate the effectiveness of using the linear kernel for dual functions, we design a linear kernel variant and test it in the super hard scenarios of SMAC. The results are shown in Table 6 .We note that using the linear kernel to parameterize dual functions leads to significant performance decline. We suspect this is because the dual function may not be linear functions. Using the linear kernel constraints the representation ability of the dual function.

H Evaluations of different values for the weight of the intrinsic reward α

The values for the weight of the intrinsic reward α in different scenarios are listed in Table 5 in our paper. To investigate the effect of different weights of the intrinsic reward, we evaluate different weight values in the easy scenario 3s5z and the super hard scenario corridor. The results are shown in Table 7. The results demonstrate that our method is not very sensitive to the values of the weight. Sub-optimal weights do not result in a significant performance drop even in the super hard scenario.

Table 6: Performance comparisons of WCD with different kernel functions in the scenarios of SMAC

Method	6h_vs_8z	corridor	3s5z_vs_3s6z
WCD (Linear Kernel)	0.57 ± 0.07	0.39 ± 0.05	0.32 ± 0.03
WCD (Ours)	0.85 ± 0.03	0.90 ± 0.03	0.87 ± 0.04

Table 7: Performance comparisons of WCD with different values for the weight of the intrinsic reward α.

Method	3s5z			corridor		
	$\alpha = 0.02$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.02$	$\alpha = 0.05$	$\alpha = 0.1$
WCD	0.89 ± 0.03	0.91 ± 0.02	0.93 ± 0.03	0.82 ± 0.07	0.85 ± 0.04	0.81 ± 0.05

I Evaluations of different cost functions

In our paper, we mainly use the Wasserstein distance to encourage sufficient exploration and simply adopt the Euclidean distance as the cost function as in many prior works. We may also use cosine similarity as the cost function, which measures the direciton differences between data points. We test the cosine similarity in Pac-Men, where agents need to move to different directions. The resutls are shown in Table 8. We note that the Wasserstein distance based on the cosine similarity achieves higher rewards in Pac-Men. In our work, we do not specifically discuss different cost functions and use the default Euclidean distance because we want to be consistent with prior works using the Wasserstein distance to ensure a fair comparison.

Table 8: Performance comparisons of WCD using different cost functions.

Method	Pac-Men
WCD (Cosine Similarity)	94 ± 0.05
WCD (Euclidean Distance)	87 ± 0.03

J Additional computational overhead analysis

We note that compared to QMIX, our method needs to train the trajectory encoder by minimizing the InfoNCE loss and the dual functions to calculate the Wasserstein distance objective, which brings additional computational overhead. We next present the comparisons of training time between QMIX and our method in the three super hard scenarios of SMAC under the same computation platform in Table 9. The results show that compared to QMIX, the addition of our proposed method does not consume a lot of extra training time.

K Training Details and Hyperparameters

In this section, we provide the training details and hyperparameters adopted in our experiments. To implement the one-step prediction method, we use a two-layer MLP with a hidden size of 64 for the encoder g_{θ_e} followed by the batch normalization and a GRU unit for the autoregressive model g_{θ_g}. We adopt a dual vector with a dimension m of 64 to parameterize the dual function. To integrate our method with QMIX, the intrinsic agent utility network is implemented with a two-layer MLP with a hidden size of 64. We keep other components the same as in QMIX.

The policy networks of all agents are implemented with Deep Recurrent Q-Networks. At each time step, an agent's policy network processes a local observation as input, which is then forwarded through a fully-connected hidden layer, followed by a GRU unit, and ultimately a fully-connected layer generating U outputs, where U is the number of actions. Furthermore, all agents' policies share the same policy network parameters to accelerate training. We set the evaluation interval to 10K steps followed by 32 test episodes. We run all methods for 5 million steps in all tested tasks. We employ hard updates to update target networks every 200 episodes in SMAC and SMACv2. In Pac-Men, we utilize soft updates for updating target networks with a momentum of 0.01. The common hyperparameters are consistent across various methods for each multi-agent task. Detailed

Table 9: Comparisons of training time between QMIX and our method in the three super hard scenarios of SMAC

Methods	Training time		
	6h_vs_8z	corridor	3s5z_vs_3s6z
QMIX	8h 35m 29s	7h 17m 30s	10h 42m 18s
WCD+QMIX	8h 43m 21s	7h 36m 53s	10h 49m 37s

hyperparameters are provided in Table 10. The replay buffer size is set to 5K. We implement our method using NumPy and PyTorch. All experiments are performed on a NVIDIA GeForce RTX 4090 GPU.

Table 10: Hyperparameters

	Pac-Men	**SMAC**	**SMACv2**
hidden dimension	64	128	
learning rate	0.0003	0.005	
optimizer		Adam	
target update	0.01(soft)	200(hard)	
batch size	32	64	
β	0.03	0.05	
α for WCD+QMIX	0.01	0.005 for 3s5z, 2c_vs_64zg, 8m, 5m_vs_6m, 8m_vs_9m, and 10m_vs_11m, 0.05 for 7sz, 6h_vs_8z, corridor, and 3s5z_vs_3s6z	0.03
α for WCD+MAPPO	0.01	0.005 for 3s5z, 2c_vs_64zg, 8m, 5m_vs_6m, 8m_vs_9m, and 10m_vs_11m, 0.03 for 7sz, 6h_vs_8z, corridor, and 3s5z_vs_3s6z	0.03
epsilon anneal time	200,000	200,000 for 3s5z, 2c_vs_64zg, 8m, 5m_vs_6m, 8m_vs_9m, and 10m_vs_11m, 500,000 for 7sz, 6h_vs_8z, corridor, and 3s5z_vs_3s6z	500,000

L Visualizations

Challenging tasks typically necessitate complex cooperative behaviors requring agents to learn diverse policies. We next present some visualization examples of diverse policies learned by our method in the super hard scenarios (6h_vs_8z, corridor, and 3s5z_vs_3s6z) in Figure 7. In the 6h_vs_8z scenario, one agent first leaves the team, causing most enemies to follow the lone agent's movements. The agent continues moving away to draw the enemies' fire and cover other agents. Other agents then quickly surround the few remaining enemies. Through learning such cooperative tactics, agents successfully scatter the enemies' powerful attacks. If all agents behave similarly and directly move towards enemies, they would be killed by enemies immediately. Similar tactics can also be observed in the other two scenarios, demonstrating the effectiveness of our method in encouraging multi-agent diversity.

We also present the visitation heatmaps of baseline methods and our method in the protoss_5_vs_5 and the zerg_5_vs_5 scenarios in Figures 8 and 9, respectively. The visitation heatmaps reveal that our proposed WCD leads to more sufficient exploration compared to the baselines. We believe this is because the baseline methods does not provide effective incentives for exploring the environment. As a result, the agents trained by baseline methods are slow to search for randomly appearing enemies on the map. In contrast, our method enables sufficient exploration by enlarging the Wasserstein distance.

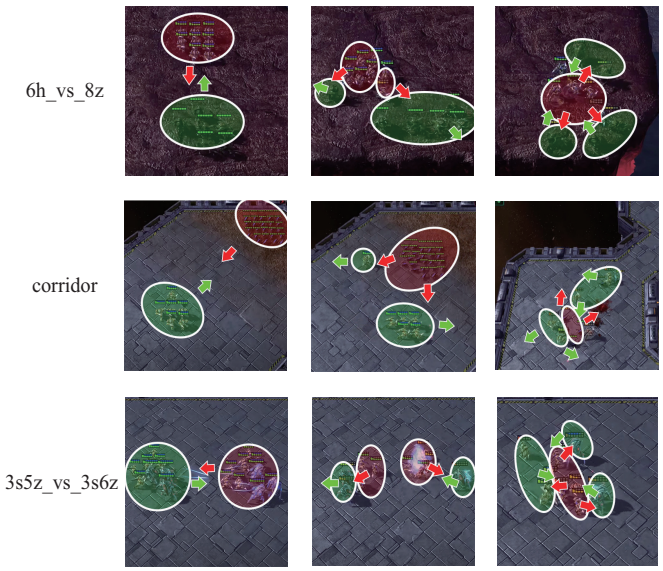
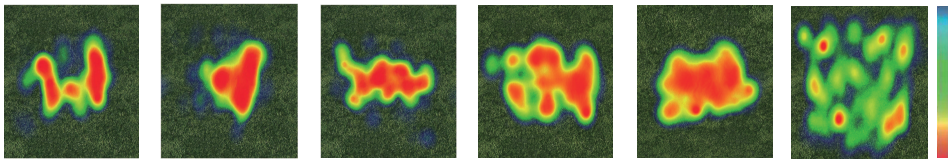
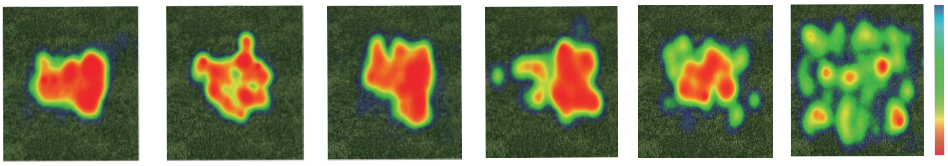


Figure 7: Visualization examples of diverse policies emerging in 6h_vs_8z (top), corridor (medium), and 3s5z_vs_3s6z (bottom) from initial (left) to final (right). Green and red shadows represent agents and enemies, respectively. Green and red arrows represent the moving directions of agents and enemies, respectively.



(a) MAVEN (b) EOI (c)SCDS (d) LIPO (e) MAPD (f) WCD

Figure 8: Visitation heatmaps of different algorithms in the protoss_5_vs_5 scenario.



(a) MAVEN (b) EOI (c)SCDS (d) LIPO (e) MAPD (f) WCD

Figure 9: Visitation heatmaps of different algorithms in the zerg_5_vs_5 scenario.