Gradient-Protected Value Decomposition for Cooperative Multi-Agent Reinforcement Learning

Jie Hou, Haowen Dou, Lujuan Dang*, Liangjun Chen, Chenyang Ge

State Key Laboratory of Human-Machine Hybrid Augmented Intelligence
National Engineering Research Center for Visual Information and Applications
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, China, 710049
{houjie, douhaowen}@stu.xjtu.edu.cn, {danglj, liangjunchen}@xjtu.edu.cn, cyge@mail.xjtu.edu.cn

Abstract

In recent years, deep multi-agent reinforcement learning (MARL) has demonstrated remarkable potential in solving complex cooperative tasks by enabling decentralized yet efficient coordination among agents. However, during decentralized training, agent policy updates induced by different joint action samples may conflict, leading to gradient interference that hinders convergence and the emergence of coordinated behavior. In this paper, we analyze and empirically validate the phenomenon of gradient interference. To address this, we then propose Gradient-Protected Value Decomposition (GPVD), a novel MARL framework that explicitly protects the gradient signals of optimal collaborative actions by suppressing the impact of interfering actions. GPVD employs a dynamic gradient protection mechanism that identifies optimal collaborative joint actions and reweights the loss to attenuate gradients from non-collaborative interfering actions. To effectively identify high-value collaborative actions, we apply SimHash-based state grouping to discover consistent collaboration patterns across similar states. Furthermore, a count-based intrinsic reward is incorporated to encourage exploration and improve the coverage of potentially optimal joint actions. Experiments on challenging multi-agent benchmarks demonstrate that GPVD achieves faster convergence, stronger coordination, and greater training stability compared to state-of-the-art value decomposition methods.

Introduction

Multi-agent reinforcement learning (MARL) has great promise to solve many real-world multi-agent problems, such as autonomous cars (Schmidt et al. 2022; Han et al. 2022), energy networks (Wang et al. 2021) and robotics (Matignon et al. 2012). The fully cooperative multi-agent task is the most common scenario, where all agents must cooperate to achieve the same goal (Lowe et al. 2017). MARL enables effective cooperation by training multiple agents together to maximize overall team performance. However, developing efficient cooperative policies remains a substantial challenge due to partial observability, non-stationary environments and the need for scalability (Oliehoek, Amato et al. 2016; Hernandez-Leal, Kartal, and Taylor 2019; Yang et al. 2020). A commonly adopted paradigm to address these issues is the Centralized Training with Decentralized Execution (CTDE) framework (Foerster et al. 2016; Sunehag et al. 2017; Rashid et al. 2020b). Under CTDE, agents are trained with access to global state in a centralized manner, but execute policies based only on local observations, thereby promoting scalability, robustness, and generalization during decentralized execution. Most MARL algorithms adopt this framework, including both policy-based approaches (Lowe et al. 2017; Yu et al. 2022; Foerster et al. 2018) and value decomposition methods (Sunehag et al. 2017; Son et al. 2019; Shen et al. 2022; Xu et al. 2023). Among them, value decomposition methods have gained significant popularity for their favorable scalability and sample efficiency in off-policy settings (Rashid et al. 2020b; Wang et al. 2020; Li et al. 2024b), and have achieved state-of-the-art (SOTA) performance on the StarCraft II Multi-Agent Challenge (SMAC) benchmark (Samvelyan et al. 2019).

The core idea of value decomposition methods is to represent the joint state-action value using neural networks as a function of individual utility functions. Most approaches adopt a mixing function that satisfies the Individual-Global-Max (IGM) principle (Son et al. 2019), enabling decentralized agents to independently select actions based on individual utility functions that are consistent with maximizing the joint value. Here, the individual utility serves as a proxy for each agent's Q-function, and agents follow greedy policies by choosing actions that maximize their own utility. To enforce this property, methods such as VDN (Sunehag et al. 2017) and QMIX (Rashid et al. 2020b) impose strict monotonicity constraints, ensuring that the joint value is a non-decreasing function of each agent's utility during training. Even expressive methods like QPLEX (Wang et al. 2020) apply non-negativity constraints to parts of the mixing network, inducing a non-negative gradient flow from the joint value to individual utilities throughout the training process. The alignment between the optimization directions of the joint value and individual utilities facilitates credit assignment and suits cooperative MARL settings, where improvements in individual utilities are assumed to monotonically increase the global team value (Hu et al. 2021). However, this structural constraint also introduces a significant drawback. In this work, we show that the constraint of non-negative gradient flow during training can induce

*Corresponding author.

gradient interference, whereby training signals from non-collaborative joint actions conflict with those from optimal collaborative joint actions, thereby hindering effective policy optimization. Under value decomposition frameworks, each agent's utility function directly determines its greedy policy. The optimization of individual utilities for optimal collaborative joint actions can be adversely affected by related joint actions, where only a subset of agents act optimally. For example, when agents coordinate to focus fire, a teammate recklessly rushing into the enemy may introduce conflicting gradients that disrupt the learning of such cooperative tactics (Mahajan et al. 2019; Kim and Sung 2023; Rashid et al. 2020a). This interference allows non-collaborative behaviors to dominate the update process, suppressing informative signals from truly optimal actions.

In this paper, we theoretically and empirically analyze the phenomenon of gradient interference in value decomposition methods and propose Gradient-Protected Value Decomposition (GPVD) framework to address this issue. GPVD first clusters similar states using SimHash-based state grouping, enabling effectively identification of high-value collaborative actions. It then introduces a dynamic gradient protection mechanism to overcome gradient interference by suppressing the impact of interfering actions. Specifically, this mechanism can identifies optimal collaborative joint actions and selectively downweights conflicting gradients from non-collaborative behaviors through loss reweighting. To further enhance exploration, GPVD incorporates a count-based intrinsic reward that encourages agents to visit under-explored states and discover potential collaborative actions. GPVD promotes efficient learning and exploration of optimal collaborative strategies, leading to stronger performance across cooperative MARL benchmarks.

Related Works

Value Decomposition Approaches Value function decomposition is a key paradigm in cooperative MARL, where a centralized action-value function is factorized into individual utilities. Methods like VDN (Sunehag et al. 2017) and QMIX (Rashid et al. 2020b) follow the Individual-Global-Max (IGM) principle (Son et al. 2019), enabling decentralized execution by aligning globally optimal joint actions with individually greedy ones. VDN uses additive factorization, while QMIX enforces monotonicity via a non-negative mixing network. Although IGM principle enables decentralized execution, the monotonicity constraint severely limits the expressiveness of Q_{tot}, making it difficult to model complex, non-monotonic agent interactions. To address this, methods such as QTRAN (Son et al. 2019), QPLEX (Wang et al. 2020), and ResQ (Shen et al. 2022) aim to relax this constraint, enhancing expressiveness by incorporating joint action information or residual learning. However, these methods often suffer from optimization challenges due to inequality constraints and reliance on precise identification of optimal joint actions.

Weighted Value Decomposition Another line of work improves value decomposition by incorporating weighted training objectives to emphasize high-value joint actions. QMIX (Rashid et al. 2020b) learns a monotonic joint action-value function by projecting Q_{tot} into a restricted function class $\mathcal{Q}_{\text{mix}}$ through supervised regression:

$$\arg\min_{q\in\mathcal{Q}_{\text{mix}}} \sum_{\boldsymbol{a}\in\boldsymbol{A}} \left(\mathcal{T}^* Q_{\text{tot}}(s,\boldsymbol{a}) - q(s,\boldsymbol{a})\right)^2,$$

where $\mathcal{T}^*$ is Bellman optimality operator. To prioritize optimal joint actions during training, weighted QMIX (Rashid et al. 2020a) introduces a weighting function $w(s,\boldsymbol{a})$ into the projection loss:

$$\arg\min_{q\in\mathcal{Q}_{\text{mix}}} \sum_{\boldsymbol{a}\in\boldsymbol{A}} w(s,\boldsymbol{a}) \left(Q(s,\boldsymbol{a}) - q(s,\boldsymbol{a})\right)^2,$$

where $Q(s,a)$ denotes the target joint action-value, $w(s,\boldsymbol{a}) = 1$ if $\boldsymbol{u}$ is optimal joint action, else $\alpha, \alpha \in (0,1]$. Similarly, QMIX-OVI (Li et al. 2024a) introduces multiple optimistic instructors to more accurately identify optimal joint actions, thereby guiding the learning process. POWQMIX (Huang et al. 2024), on the other hand, learns a complete IGM function, selects the joint action with the highest predicted value, and assigns lower weights to all remaining actions.

Despite their effectiveness, these methods heavily rely on accurate identification of the optimal joint action at each individual state, while failing to exploit structural similarities across states (Jianye et al. 2022; Na and Moon 2024). Moreover, assigning near-zero weights to all non-optimal joint actions based on such approximations significantly limits the contribution of many training samples, reducing sample efficiency. In contrast, we leverage SimHash-based state grouping to capture consistent collaboration patterns and selectively downweight interfering samples, thereby both enhancing the learning of optimal collaborative actions and preserving sample efficiency.

Preliminaries

Decentralized POMDP A fully cooperative multi-agent task where agents make decisions in a decentralized setting is usually modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek, Amato et al. 2016), described by the tuple $M = \langle \mathcal{N}, \mathcal{S}, \boldsymbol{A}, \mathcal{R}, \mathcal{P}, \boldsymbol{Z}, \boldsymbol{O}, \gamma \rangle$, where $\mathcal{N} = \{1, 2, \ldots, n\}$ denotes the agent set and S is the state space. $\boldsymbol{A} = \left\{A^1 \times A^2 \ldots \times A^n\right\}$ represents the joint action space of all agents. At each time step t, each agent i receives its local observation $o_t^i \in Z^i \in \boldsymbol{Z}$ according to its observation function $O^i\left(o_t^i \mid s_t\right) \in \boldsymbol{O}$ and selects its local action $a_t^i \in A^i$. After all agents select actions, the environment transits to the next state s_{t+1} according to the state transition function $\mathcal{P}\left(s_{t+1} \mid s_t, \boldsymbol{a_t}\right)$ and provides the global reward r_t according to the reward function $\mathcal{R}\left(s_t, \boldsymbol{a_t}\right)$, where $\boldsymbol{a_t} = \left(a_t^1, a_t^2, \ldots, a_t^n\right)$ denotes the joint action of all agents. γ is a discount factor. To overcome the partially observability challenge, each agent i executes a decentralized policy $\pi^i\left(a_t^i \mid \tau_t^i\right)$, where $\tau_t^i = \left(o_0^i, a_0^i, o_1^i, a_1^i, \ldots, o_t^i\right)$ denotes its local action-observation history. Then, the joint policy $\boldsymbol{\pi}$ is given as the product form $\boldsymbol{\pi} = \prod_{i=1}^n \pi^i$. The goal of all agents is to learn the optimal joint policy $\boldsymbol{\pi}^* =$

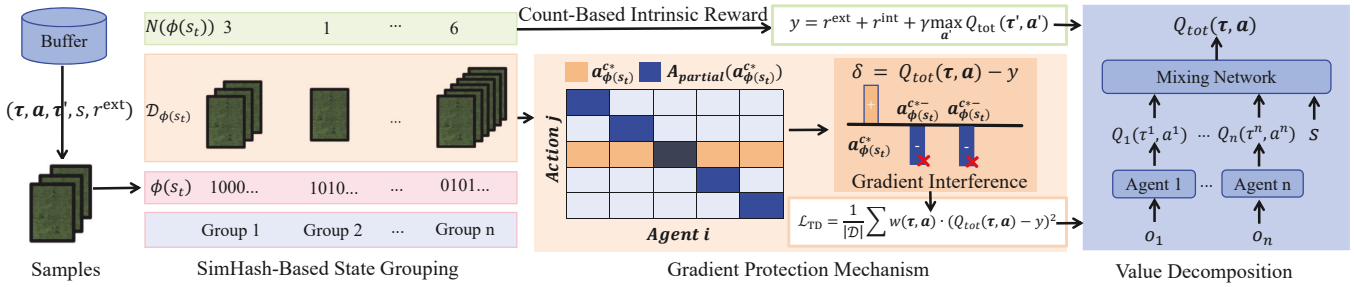


Figure 1: The overall architecture of GPVD. GPVD includes three core components: (1) SimHash-based state grouping that clusters similar states to promote identification of collaborative behaviors; (2) Gradient protection mechanism that identifies and protects optimal collaborative joint actions; and (3) Count-based intrinsic reward to further encourage exploration.

$\{\pi^{1,*}, \pi^{2,*}, \ldots, \pi^{n,*}\}$ that maximizes the cumulative discounted rewards $\mathbb{E}_{\boldsymbol{\pi},\mathcal{P}}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.

Non-Negative Gradient Flow In value function decomposition, the Individual-Global-Max (IGM) principle (Son et al. 2019) is introduced to ensure consistency between the global joint value function and individual agent utilities. It is formally defined as follows: $\arg\max_{\boldsymbol{a}} Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a}) = \prod_{i=1}^{n} \arg\max_{a^i} Q_i(\tau^i, a^i)$, where $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_n)$ is the joint trajectory. This ensures that independently greedy actions collectively yield the globally optimal joint action. To guarantee such consistency, most value decomposition algorithms construct the global value function to be monotonic with respect to the individual utility functions, typically by enforcing non-negative mixing weights (Li et al. 2024b; Xu et al. 2023). This structural constraint ensures that the gradient flow with respect to each utility remains non-negative during training:

$$\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0, \quad \forall i \in \mathcal{N}. \tag{1}$$

In more expressive models like QPLEX, the global value function is defined as $Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=1}^{n} \left(Q_i(\boldsymbol{\tau}, a^i) + (\lambda_i - 1) Adv_i(\boldsymbol{\tau}, a^i)\right)$, where Adv_i is the advantage function and $\lambda_i \geq 0$ is a learnable weight possibly dependent on the global state and joint action. To stabilize training, gradient from the advantage term Adv_i is detached during backpropagation (Wang et al. 2020). As a result, although Q_{tot} is not structurally monotonic, the gradient flow with respect to each Q_i remains non-negative during optimization. Therefore, *non-negative gradient flow* is a more general property than the monotonicity constraint in value decomposition. It ensures that, during training, the gradients of all individual utilities Q_i corresponding to a joint action align with the optimization direction of Q_{tot}. Consequently, the update direction for $Q_i(a^i)$ is influenced by all joint actions that include the individual action a^i.

Methodology

We begin by analyzing the gradient signals of individual utilities in value decomposition and formally define *Gradient Interference*, demonstrating its impact on policy learning via a matrix game. We then introduce our proposed algorithm, *Gradient-Protected Value Decomposition* (GPVD), which effectively overcome this issue. Figure 1 shows the architecture of our learning framework.

Gradient Interference Existing value decomposition methods often suffer from gradient interference, where conflicting updates from different joint actions hinder the optimization of individual utilities and disrupt policy learning. To better understand and address this issue, we draw inspiration from counterfactual reasoning and define collaborative and non-collaborative joint actions based on local deviation analysis over the joint value function. Specifically, we are interested in joint actions that outperform all alternatives that differ in some agents' actions while sharing at least one agent's action. So, we define a neighborhood set of joint actions that are partially aligned with a collaborative joint action $\boldsymbol{a}_c$: $\boldsymbol{A}_{\text{partial}}(\boldsymbol{a}_c) = \left\{\boldsymbol{a} \in \boldsymbol{A} \,\middle|\, \boldsymbol{a} \neq \boldsymbol{a}_c \text{ and } \exists i \in \mathcal{N},\ a^i = a^i_c\right\}$.

Definition 1 (Collaborative and Non-Collaborative Joint Actions)**.** *A joint action $\boldsymbol{a}_c$ is called a collaborative joint action at state s if it satisfies:*

$$Q_{jt}(s, \boldsymbol{a}_c) \geq Q_{jt}(s, \boldsymbol{a}), \quad \forall \boldsymbol{a} \in \boldsymbol{A}_{partial}(\boldsymbol{a}_c), \tag{2}$$

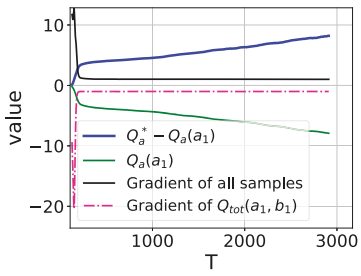
and the related non-collaborative joint action $\boldsymbol{a}_{c-} \in \boldsymbol{A}_{partial}(\boldsymbol{a}_c)$ that satisfies: $Q_{jt}(s, \boldsymbol{a}_{c-}) < Q_{jt}(s, \boldsymbol{a}_c)$. Here, $Q_{jt}(s, \boldsymbol{a})$ denotes the true joint action-value function.

Among all collaborative joint actions, we consider the *optimal collaborative joint action* $\boldsymbol{a}^*_c$ as the one that achieves the highest joint value: $\boldsymbol{a}^*_c = \arg\max_{\boldsymbol{a}_c} Q_{\text{jt}}(s, \boldsymbol{a}_c)$. Thus, effective policy learning requires protecting the gradient signals of the optimal collaborative action $\boldsymbol{a}^*_c$ while suppressing interference from conflicting non-collaborative actions $\boldsymbol{a}^*_{c-}$. In value decomposition framework, $Q_{\text{jt}}(s, \boldsymbol{a})$ is approximated by a decomposable total value function $Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a})$. For each transition sample, we can analyze the gradient of the loss with respect to individual utilities. For a given state s, suppose we collect some transition samples indexed by k. For each sample k, the temporal-difference (TD) error is defined as $\delta_k = Q_{\text{tot}}(\boldsymbol{\tau}_k, \boldsymbol{a}_k) - y_k$, where y_k is the corresponding TD target. Then the TD loss is then computed as $\mathcal{L}^k_{\text{TD}} = \frac{1}{2}\delta^2_k$. Due to the non-negative gradient flow, the gradient of the TD loss with respect to each individual utility Q_i for sample k is given by:

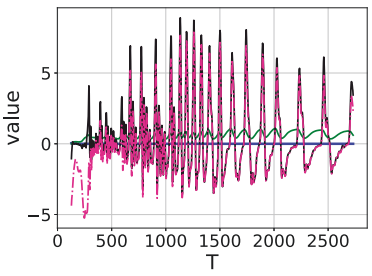
$$\nabla_{Q_i} \mathcal{L}^k_{\text{TD}} = \delta_k \cdot \frac{\partial Q_{\text{tot}}}{\partial Q_i} \propto \delta_k, \quad \forall i \in \mathcal{N}. \tag{3}$$

B \ A	a_1	a_2	a_3
b_1	**8**	-12	-12
b_2	-12	**6**	0
b_3	-12	0	**6**

(a) Payoff matrix



(b)



(c)

Figure 2: Gradient interference in the payoff matrix scenario. (a) The payoff matrix. (b) Gradient interference on $Q_a(a_1)$ induced by the non-collaborative samples (a_1, b_2) and (a_1, b_3). (c) show the training dynamics with gradient protection applied.

Then, we define the Gradient Interference as follows:

Definition 2 (Gradient Interference)**.** *Let δ_c and δ_{c-} denote the TD errors of a collaborative joint action $\boldsymbol{a}_c$ and a non-collaborative joint action $\boldsymbol{a}_{c-}$, respectively. Gradient interference on Q_i occurs if*

$$\nabla_{Q_i}\mathcal{L}_{\mathrm{TD}}^{c} \cdot \nabla_{Q_i}\mathcal{L}_{\mathrm{TD}}^{c-} \propto \delta_c \cdot \delta_{c-} < 0, \quad \forall i \in \{i \mid a_c^i = a_{c-}^i\}, \tag{4}$$

indicating that conflicting TD errors induce opposing gradient directions on shared utility components, thereby disrupting optimal policy learning.

Gradient interference's negative impact on policy learning can be illustrated by a simple matrix game, as shown in the Figure 2. In this scenario, agent A and agent B execute a joint action to receive a reward specified in the payoff table. Q_a and Q_b denote the utility functions of agent A and B, and Q^* represent the optimal value. The joint actions $(a_1, b_1), (a_2, b_2), (a_3, b_3)$ are collaborative joint actions, with (a_1, b_1) being the optimal collaborative joint action. Figure 2(b) illustrates the evolution of utility function outputs and the corresponding gradient flows during training. $Q_a(a_1)$ closely follow the aggregated gradient directions from all samples: the utilities increase when the gradient is negative and decrease when it is positive. This phenomenon is further highlighted in Figures 2(c). However, the beneficial gradient signal from the optimal joint action (a_1, b_1) is overwhelmed by conflicting updates induced by non-collaborative joint actions. As a result, $Q_a(a_1)$ deviate from Q_a^*, causing the policy to converge to a suboptimal solution. A similar phenomenon is observed for $Q_b(b_1)$.

SimHash-Based State Grouping To protect the gradient signals of optimal collaborative joint actions during training, it is essential to accurately identify such actions at each state. To effectively identify optimal collaborative joint actions in multi-agent environments with high-dimensional and continuous state spaces (Yang et al. 2020; Zheng et al. 2021; Yu et al. 2022), we employ SimHash as a lightweight yet efficient state encoder to discretize the state space by mapping semantically similar states into the same hash bucket (Charikar 2002; Tang et al. 2017). This discretization enables efficient local action analysis and facilitates sample-efficient identification of high-return collaborative actions.

Let $s \in \mathcal{S}$ denote a high-dimensional continuous state. SimHash projects s into a k-bit binary code by applying a randomly initialized projection matrix $\mathbf{A} \in \mathbb{R}^{k \times D}$:

$$\phi(s) = \mathrm{sign}(\mathbf{A}g(s)) = [\mathbb{I}(\mathbf{a}_1 g(s) \geq 0), \ldots, \mathbb{I}(\mathbf{a}_k g(s) \geq 0)], \tag{5}$$

where $g : \mathcal{S} \to \mathbb{R}^D$ is an optional preprocessing function and $\mathbf{a}_i$ is the i-th row of $\mathbf{A}$, sampled from a standard Gaussian distribution, $\mathbb{I}(\cdot)$ is the indicator function. The value for k controls the granularity: higher values lead to fewer collisions and are thus more likely to distinguish states. For any given query state s_t, we define the corresponding sample group (or sample bucket) as:

$$\mathcal{D}_{\phi(s_t)} = \left\{(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{\tau}', s, r^{\mathrm{ext}}) \in \mathcal{D} \,\middle|\, \phi(s) = \phi(s_t)\right\}, \tag{6}$$

where $\mathcal{D}$ denotes the dataset of transition tuples $(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{\tau}', s, r^{\mathrm{ext}})$ collected during training, and r^{ext} represents the extrinsic rewards obtained from the environment. The grouped sample set $\mathcal{D}_{\phi(s_t)}$ serves as the statistical basis for identifying consistent collaborative joint actions that are not specific to a single state instance, but rather generalize over a local neighborhood in the state space. This design maps states into a unified semantic space, allowing for the identification of consistent collaborative structures across similar states, which is essential for the gradient protection mechanism introduced next.

Gradient Protection Mechanism To mitigate the impact of gradient interference, we introduce a mechanism that protects collaborative joint actions by reweighting the loss function. Specifically, the contributions of actions exhibiting gradient conflict are down-weighted to reduce their adverse effect on policy optimization, while the gradients of optimal collaborative actions are preserved to guide effective coordination learning. However, during policy optimization, the true optimal collaborative actions are not directly observable. In practice, we identify actions requiring gradient protection by selecting those with the highest TD targets within each state bucket. And define the optimal collaborative joint action in group $\mathcal{D}_{\phi(s_t)}$ as:

$$\boldsymbol{a}_{\phi(s_t)}^{c*} := \underset{\boldsymbol{a} \in \boldsymbol{A}_{\phi(s_t)}}{\arg\max} \left\{ y(\boldsymbol{\tau}, \boldsymbol{a}) \mid y(\boldsymbol{\tau}, \boldsymbol{a}) > \max_{\boldsymbol{a}'} Q_{\mathrm{tot}}(\boldsymbol{\tau}, \boldsymbol{a}') \right\},$$

where $\boldsymbol{A}_{\phi(s_t)} = \left\{\boldsymbol{a} \mid (\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{\tau}', s, r^{\text{ext}}) \in \mathcal{D}_{\phi(s_t)}\right\}$ is the set of joint actions observed within the group, and $y(\boldsymbol{\tau}, \boldsymbol{a}) = r(\boldsymbol{\tau}, \boldsymbol{a}) + \gamma \max_{\boldsymbol{a}'} Q_{\text{tot}}(\boldsymbol{\tau}', \boldsymbol{a}')$ denotes the corresponding TD target. Then, $\boldsymbol{a}^{c*}_{\phi(s_t)}$ serves as a high-return, underfitted action for collaborative policy optimization. This ensures that the selected action not only demonstrates a high estimated return, but also exceeds the current maximum Q-value under the learned policy, thereby identifying potentially overlooked optimal collaborative joint action. State grouping is computed per round to update counts and identify potential optimal actions, which has minimal impact on the method's scalability.

To protect the gradient signals of the optimal collaborative joint action $\boldsymbol{a}^{c*}_{\phi(s_t)}$, it is crucial to suppress the influence of its associated non-collaborative actions. To maximize sample efficiency, it is not necessary to suppress all non-collaborative actions. As previously analyzed (Eq. 4), only those non-collaborative joint actions whose TD error signs conflict with that of the optimal collaborative joint action should be attenuated. Specifically, with respect to $\boldsymbol{a}^{c*}_{\phi(s_t)}$, we denote the set of non-collaborative joint actions that cause gradient interference as:

$$\boldsymbol{a}^{c*-}_{\phi(s_t)} = \left\{\boldsymbol{a} \in \boldsymbol{A}_{\text{partial}}(\boldsymbol{a}^{c*}_{\phi(s_t)}) \text{ and } \delta(\boldsymbol{a}) \cdot \delta(\boldsymbol{a}^{c*}_{\phi(s_t)}) < 0\right\}.$$

where $\delta(\cdot)$ denotes the TD error and $\boldsymbol{a} \in \boldsymbol{A}_{\phi(s_t)}$. Then, we introduce a sample-wise weighting function $w(\boldsymbol{\tau}, \boldsymbol{a})$ that adaptively adjusts the contribution of each transition to the overall loss:

$$w(\boldsymbol{\tau}, \boldsymbol{a}) = \begin{cases} \alpha, & \text{if } \boldsymbol{a} \in \boldsymbol{a}^{c*-}_{\phi(s_t)}, \\ 1, & \text{otherwise}, \end{cases} \tag{7}$$

where $\alpha \in (0, 1)$ is a fixed small factor that down-weights conflicting samples. To further accelerate convergence in cases where optimal collaborative actions are under-sampled, we may also assign a larger weight to the transitions associated with $\boldsymbol{a}^{c*}_{\phi(s_t)}$, thereby amplifying their gradient signal. Based on this weighting scheme, the final TD loss objective becomes:

$$\mathcal{L}_{\text{TD}} = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{\tau}', s, r^{\text{ext}}) \in \mathcal{D}} w(\boldsymbol{\tau}, \boldsymbol{a}) \cdot \left(Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a}) - y\right)^2, \tag{8}$$

This loss formulation ensures that updates from non-collaborative joint actions with conflicting TD directions are suppressed during training.

Count-Based Intrinsic Reward To further improve exploration and prevent premature convergence, we integrate a count-based intrinsic reward into the gradient protection framework (Jeon et al. 2022; Jo et al. 2024). This intrinsic reward encourages agents to visit less-explored state regions more frequently, thereby promoting diverse experience collection and reducing the risk of converging prematurely to suboptimal joint policies (Bellemare et al. 2016; Tang et al. 2017). Formally, for each state cluster defined by the SimHash encoding, we maintain a visitation count $N(\phi(s_t))$. The intrinsic reward is computed as:

$$r^{\text{int}}(s_t) = \frac{\beta}{\sqrt{N(\phi(s_t))}} + \kappa, \tag{9}$$

where $\beta > 0$ controls the strength of the intrinsic reward, and κ is a fixed offset term. By integrating the intrinsic reward with the extrinsic task reward, agents obtain a balanced learning signal that promotes both effective collaboration and efficient exploration. Consequently, the new TD target is defined as $y = r^{\text{ext}} + r^{\text{int}} + \gamma \max_{\boldsymbol{a}'} Q_{\text{tot}}(\boldsymbol{\tau}', \boldsymbol{a}')$,where γ is the discount factor.

B \ A	a_1(-24.49)	a_2(0.058)	a_3(-0.200)
b_1(-24.51)	-9.594	-9.569	-9.571
b_2(0.070)	-9.563	**5.434**	1.316
b_3(-0.200)	-9.565	1.319	0.843

(a) Payoff learned by QMIX

B \ A	a_1 (0.205)	a_2 (-1.838)	a_3 (-1.812)
b_1 (0.187)	**8.036**	-5.455	-5.455
b_2 (-1.846)	-5.453	-5.459	-5.459
b_3 (-1.831)	-5.453	-5.459	-5.459

(b) Payoff learned by GPVD

Figure 3: The value functions (individual utilities and total value) learned by QMIX and GPVD in Payoff Matrix.

Experiment Results

In this section, we evaluate the performance of our proposed method across three cooperative multi-agent environments: the Matrix Game, Predator-Prey, and the StarCraft II Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019). To ensure fair and consistent comparisons, all competing algorithms are trained using the same optimizer configurations and identical hyperparameter settings. All results are obtained from 5 runs under different random seeds and are plotted using means and standard deviation. Our implementation is built upon the PyMARL2 framework (Hu et al. 2021), with QMIX serving as the baseline. We integrate our method into QMIX and evaluate its performance.

Matrix Game To evaluate the effectiveness of GPVD in mitigating gradient interference, we first conduct experiments on the Matrix Game. Figure 2(c) illustrates the evolution of the utility function output and the corresponding gradient flow for the optimal joint action (a_1, b_1) during GPVD training. As shown in the figure, the gradient signal of (a_1, b_1) is successfully protected and not overwhelmed by non-collaborative actions, thanks to our proposed gradient protection mechanism. As a result, the utility value of the optimal collaborative action rapidly increases and converges within a few training steps, leading to the emergence of the optimal policy. This demonstrates that GPVD effectively identifies and exploits optimal joint behaviors. The value functions (individual utilities and total value) learned by QMIX and GPVD are provided in Figure 3.

Predator-Prey Predator-Prey is a partially observable environment where eight predators cooperate to capture eight preys on a 10×10 grid. A successful capture by two or more adjacent predators yields a shared reward of $r = 10$, while solo attempts incur a penalty $p \leq 0$. We evaluate two settings: moderate coordination ($p = 0$) and high coordination

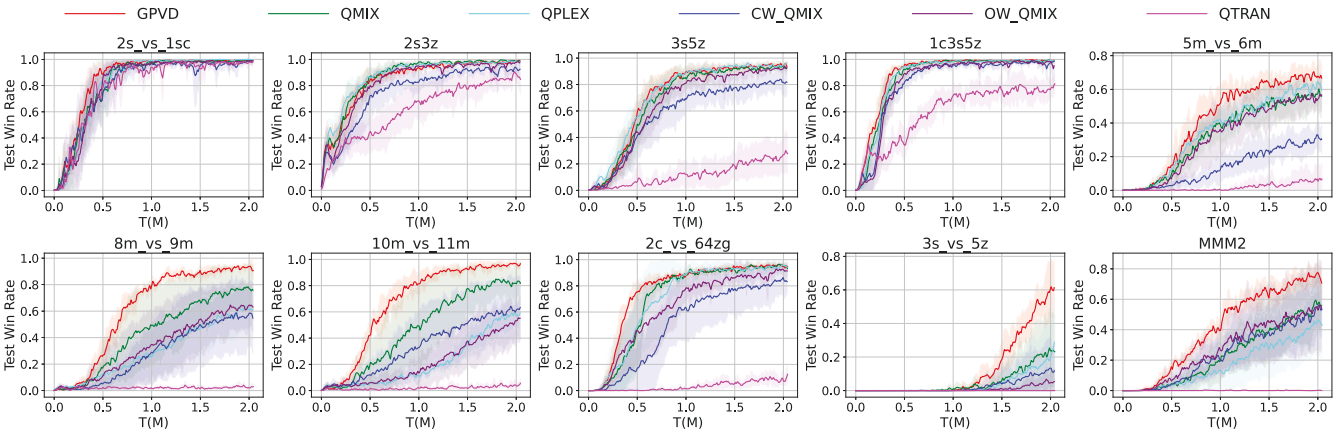


Figure 4: Test win rate on the SMAC tasks.

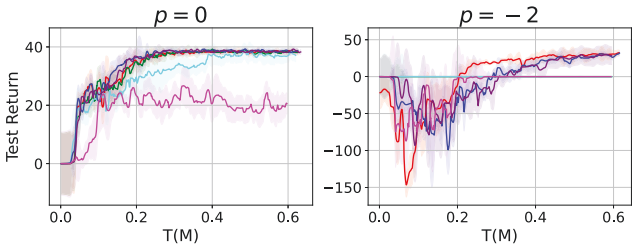


Figure 5: Test return on the Predator-Prey tasks.

Scenario	GPVD	w/o SimHash	w/o GP	w/o r^{int}
5m_vs_6m	0.67	0.52	0.63	0.61
8m_vs_9m	0.92	0.83	0.83	0.92
3s_vs_5z	0.61	0.27	0.43	0.31
MMM2	0.74	0.74	0.72	0.48

Table 1: Ablations results comparing GPVD and its ablated versions on four SMAC maps.

Hyperparameter	Value	5m_vs_6m	8m_vs_9m
α	0.1	0.70	0.92
	0.2	0.67	0.92
	0.5	0.63	0.89
	1.0	0.63	0.83
β	0.01	0.67	0.92
	0.05	0.65	0.86
	0.1	0.70	0.78
k	16	0.71	0.81
	32	0.67	0.92
	64	0.58	0.90
QMIX	-	0.58	0.76

Table 2: Hyperparameter analysis of GPVD on the `5m_vs_6m` and `8m_vs_9m`.

($p = -2$). In the latter, we emphasize optimal collaborative actions by increasing their gradient weights to promote faster convergence. Figure 5 shows the performance comparison among our method and the baselines. When p equals zero, most methods achieve satisfactory performance. Under the more challenging setting where $p = -2$, conventional methods such as QMIX, QPLEX, and QTRAN fail to learn effective coordination strategies for successful prey capture. In contrast, both GPVD and WQMIX demonstrate strong performance. Compared to WQMIX, our method converges faster by protecting gradients of optimal collaborative actions, enhancing learning under high coordination.

StarCraft II Multi-Agent Challenge (SMAC) Next, we further evaluate our proposed method on the challenging StarCraft II Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019), a widely-used benchmark for cooperative MARL. In SMAC, agents are divided into two teams and must coordinate with allies while competing against enemies controlled by the built-in game AI. To assess the effectiveness and generality of our approach, we conduct experiments on ten representative SMAC scenarios. Figure 4 presents the performance curves across all evaluated scenarios. Our method consistently achieves faster convergence and higher win rates across a wide range of tasks. In simple scenarios such as `2s_vs_1sc` and `2s3z`, GPVD performs comparably to existing state-of-the-art methods. In more complex tasks, it demonstrates notably faster convergence and superior win rates. Although QPLEX and QTRAN possess greater representational capacity compared to QMIX, their performance does not consistently surpass that of QMIX across all scenarios. Interestingly, while CW-QMIX and OW-QMIX perform competitively in simpler tasks, their performance drops significantly on the SMAC benchmark compared to the original QMIX. This degradation is likely due to their uniform down-weighting of all non-optimal actions, which inadvertently suppresses useful learning signals and leads to poor sample efficiency. In contrast, our method selectively suppresses only those samples that induce gradient interference with optimal collaborative actions, guided by state grouping. Simultaneously,

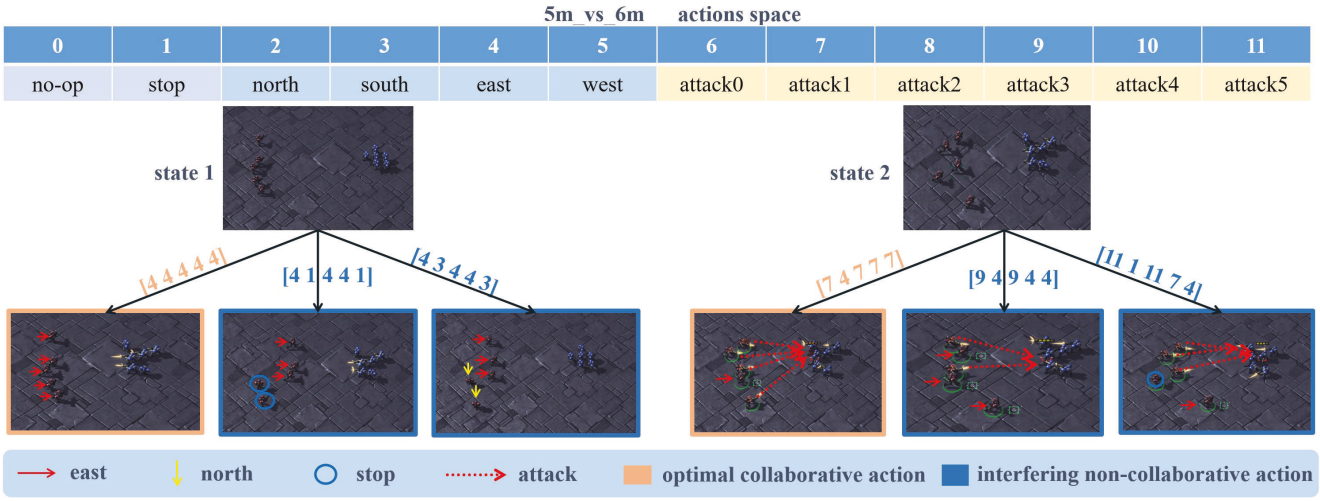


Figure 6: Visualization of optimal collaborative and interfering non-collaborative actions identified by GPVD in the 5m_vs_6m.

(a) Dispersing enemy in 3s_vs_5z

(b) Kiting in 3s_vs_5z

(c) Backline healing with frontline tanking, prioritizing high-damage and healer units first, and handling tank units last in MMM2

Figure 7: Cooperative strategies learned through GPVD.

it incorporates a count-based intrinsic reward to encourage exploration. This dual mechanism not only preserves critical learning signals essential for effective coordination, but also promotes exploration of under-visited states, thereby improving sample efficiency and enhancing the robustness of policy learning.

Ablation and Hyperparameter Analysis Table 1 presents the ablation study results. The complete GPVD model consistently outperforms its ablated variants, validating the contribution of each component. presents We further analyze the sensitivity of key hyperparameters (Table 2) and observe that GPVD exhibits good robustness across a broad range of settings. Nonetheless, achieving optimal performance still requires appropriate tuning tailored to the specific task.

Visualization To validate the effectiveness of the gradient protection mechanism, we visualize optimal collaborative joint actions and their interfering counterparts during training on 5m_vs_6m, a task that requires coordination under numerical disadvantage. Figure 6 shows two representative grouped states. To trace runtime scenarios, we visualize the episodes, where the enemy IDs are randomly assigned. In *state 1*, none of the enemies (blue) are within attack range of the allied agents (red); the optimal strategy is for all agents to move east, while GPVD correctly suppresses conflicting actions such as stopping or moving south. In *state 2*, all agents except agent 2 are positioned to attack; the optimal action is for agent 2 to move east to enter range while others focus fire. Again, GPVD effectively distinguishes two interfering joint actions that involve suboptimal behaviors, such as stopping or incorrectly repositioning. These visualizations provide intuitive evidence that gradient protection effectively identifies and suppresses conflicting updates, preserving gradients from optimal collaborative behaviors.

In harder scenarios like 3s_vs_5z and MMM2, GPVD enables rapid emergence of cooperative behaviors such as dispersing enemy, kiting, focus fire, and role-aware actions within 2 million steps (Figure 7). This demonstrates GPVD's ability to efficiently learn complex collaboration strategies.

Conclusion

In value decomposition-based multi-agent reinforcement learning, decentralized agents often suffer from gradient interference, which disrupts the learning of optimal strategies, reduces sample efficiency, and increases the risk of converging to suboptimal behaviors. To address this, we propose Gradient-Protected Value Decomposition (GPVD), a novel framework that explicitly safeguards the gradient signals of optimal collaborative actions while suppressing harmful interference from conflicting joint actions. Extensive experiments demonstrate the effectiveness and robustness of our method in complex cooperative settings.

Acknowledgments

This work is supported by the National Key R&D Program of China (2023YFB4704900) and National Natural Science Foundation of China (U21A20485).

References

Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29.

Charikar, M. S. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 380–388.

Foerster, J.; Assael, I. A.; De Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29.

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

Han, S.; Wang, H.; Su, S.; Shi, Y.; and Miao, F. 2022. Stable and efficient Shapley value-based reward reallocation for multi-agent reinforcement learning of autonomous vehicles. In *2022 International Conference on Robotics and Automation (ICRA)*, 8765–8771. IEEE.

Hernandez-Leal, P.; Kartal, B.; and Taylor, M. E. 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6): 750–797.

Hu, J.; Jiang, S.; Harding, S. A.; Wu, H.; and Liao, S.-w. 2021. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2102.03479*.

Huang, C.; Zhu, S.; Zhao, J.; Zhou, H.; Ye, C.; Feng, T.; and Jiang, C. 2024. POWQMIX: Weighted Value Factorization with Potentially Optimal Joint Actions Recognition for Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2405.08036*.

Jeon, J.; Kim, W.; Jung, W.; and Sung, Y. 2022. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International conference on machine learning*, 10041–10052. PMLR.

Jianye, H.; Hao, X.; Mao, H.; Wang, W.; Yang, Y.; Li, D.; Zheng, Y.; and Wang, Z. 2022. Boosting multiagent reinforcement learning via permutation invariant and permutation equivariant networks. In *The eleventh international conference on learning representations*.

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

Kim, W.; and Sung, Y. 2023. An adaptive entropy-regularization framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*, 16829–16852. PMLR.

Li, C.; Zhang, Y.; Wang, J.; Hu, Y.; Dong, S.; Li, W.; Lv, T.; Fan, C.; and Gao, Y. 2024a. Optimistic value instructors for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17453–17460.

Li, H.; Zhou, H.; Zou, Y.; Yu, D.; and Lan, T. 2024b. Concaveq: Non-monotonic value function factorization via concave representations in deep multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17461–17468.

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

Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. Maven: Multi-agent variational exploration. *Advances in neural information processing systems*, 32.

Matignon, L.; Jeanpierre, L.; Mouaddib, A. I.; et al. 2012. Coordinated Multi-Robot Exploration under Communication Constraints Using Decentralized Markov Decision Processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, 2017–2023.

Na, H.; and Moon, I.-c. 2024. LAGMA: Latent goal-guided multi-agent reinforcement learning. *arXiv preprint arXiv:2405.19998*.

Oliehoek, F. A.; Amato, C.; et al. 2016. *A concise introduction to decentralized POMDPs*, volume 1. Springer.

Rashid, T.; Farquhar, G.; Peng, B.; and Whiteson, S. 2020a. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10199–10210.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020b. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178): 1–51.

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

Schmidt, L. M.; Brosig, J.; Plinge, A.; Eskofier, B. M.; and Mutschler, C. 2022. An introduction to multi-agent reinforcement learning and review of its application to autonomous mobility. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 1342–1349. IEEE.

Shen, S.; Qiu, M.; Liu, J.; Liu, W.; Fu, Y.; Liu, X.; and Wang, C. 2022. Resq: A residual q function-based approach for multi-agent reinforcement learning value factorization. *Advances in Neural Information Processing Systems*, 35: 5471–5483.

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

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

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Xi Chen, O.; Duan, Y.; Schulman, J.; DeTurck, F.; and Abbeel, P. 2017. Exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2020. QPLEX: Duplex Dueling Multi-Agent Q-Learning. *arXiv e-prints*, arXiv–2008.

Wang, J.; Xu, W.; Gu, Y.; Song, W.; and Green, T. C. 2021. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in Neural Information Processing Systems*, 34: 3271–3284.

Xu, Z.; Zhang, B.; Zhou, G.; Zhang, Z.; Fan, G.; et al. 2023. Dual self-awareness value decomposition framework without individual global max for cooperative MARL. *Advances in Neural Information Processing Systems*, 36: 73898–73918.

Yang, Y.; Hao, J.; Liao, B.; Shao, K.; Chen, G.; Liu, W.; and Tang, H. 2020. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*.

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

Zheng, L.; Chen, J.; Wang, J.; He, J.; Hu, Y.; Chen, Y.; Fan, C.; Gao, Y.; and Zhang, C. 2021. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34: 3757–3769.