CAAI Transactions on Intelligence Technology

WILEY

ORIGINAL RESEARCH OPEN ACCESS

Credit-Driven Adaptive Grouping for Refined Cooperative Multi-Agent Reinforcement Learning

Yirui Liu | Xueqiang Gu | Caiming Zheng | Xin Xu | Zhangling Wang | Wanpeng Zhang | Yixing Lan

College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China

Correspondence: Wanpeng Zhang (wpzhang@nudt.edu.cn) | Yixing Lan (lanyixing16@nudt.edu.cn)

Received: 29 July 2025 | **Revised:** 31 January 2026 | **Accepted:** 1 March 2026

Keywords: grouping-based MARL | multi-agent cooperation | multi-agent reinforcement learning | policy diversity

ABSTRACT

Policy heterogeneity is crucial for achieving sophisticated coordination in complex collaborative tasks, which has emerged as one of the key challenges in multi-agent reinforcement learning (MARL) in recent years. Notably, the grouping paradigm has made remarkable progress in addressing policy heterogeneity. However, most existing grouping methods require predefining the number of groups or the composition and quantity of members within each group, which need to be individually configured for each scenario and are difficult to set without sufficient expert knowledge. By contrast, we propose a novel MARL grouping algorithm named Credit-driven adaptive Grouping (CreateG) which divides the entire training process into multiple phases and reallocates poorly adapted (low-credit) individuals at each training stage. With the help of the mechanism we designed, an environment-adaptive grouping is ultimately formed. Furthermore, we design a hierarchical hypernetwork architecture to accommodate this adaptive grouping mechanism. Experiments conducted on StarCraft II micromanagement hard and super-hard tasks, Google Research Football and TAG scenarios show CreateG achives state-of-the-art MARL performance. Moreover, extensive ablation studies elucidate the operational mechanism of the grouping strategy and other components demonstrate how they enhance overall performance.

1 | Introduction

Multi-Agent Reinforcement Learning (MARL) has emerged as a pivotal paradigm for addressing challenges that involve multiple autonomous entities interacting within dynamic environments, including coordinating traffic lights [1, 2], orchestrating multi-robot formations [3, 4] and optimising the smart grid operations [5, 6]. The framework of Centralized Training with Decentralized Execution (CTDE) [7, 8] has shown considerable advantages in mitigating the partial observation limitation and addressing scaling issues in MARL. Value Decomposition (VD) under the CTDE framework such as VDN [9], QMIX [10] and QTRAN [11] has emerged as a promising credit assignment paradigm, endowing agents with the ability to acquire their own optimal policies by decomposing the joint value function based on their individual credits. To enhance scalability and convergence, these methods adopt the parameter-sharing mechanism where all agents learn and share the same decentralised value or policy network through joint learning. However, a large body of research [12–15] has shown that this parameter-sharing mechanism may lead to representational conflicts when agents have diverse behaviours and preferences, potentially hindering the learning efficiency and the quality of the learned policies—an issue known as the lack of policy heterogeneity.

To tackle the aforementioned lack of policy heterogeneity caused by parameter sharing mechanism, existing methods can be categorised into two main aspects: identity-level and group-level methods. Identity-level methods—including EOI [16], MAVEN [17] and CIA S [5].—boost individual agent diversity through

Abbreviations: ANA, anti-nuclear antibodies; APC, antigen-presenting cells; IRF, interferon regulatory factor.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

https://doi.org/10.1049/cit2.70127

intrinsic rewards [16], latent space exploration [17], or contrastive learning [12, 18]. Yet these identity-level methods suffer from an inherent flaw: they overprioritise individual uniqueness, disrupting collective coordination as agents focus on local optimization over shared goals [19, 20]. In contrast, group-level methods present a more balanced alternative by sharing parameters within groups or roles while preserving diversity across groups [21, 22]. Specifically, RODE Tonghan [23] achieves grouping through spatial decomposition based on action effects, ACORM [24] conducts role representation learning via attention mechanisms and contrastive learning, VAST [25] adopts a value function decomposition-based variable sub-team strategy, and SOG [26] enables agents to autonomously form collaborative groups through a self-organised mechanism driven by local interactions. However, most existing grouping methods in current research are plagued by a common flaw: they all require the predefinition of either the number of groups or the composition of members within each group which need to individually tuned for every specific scenario, and their proper configuration is often hindered by the requirement for extensive expert knowledge.

Against this backdrop, we propose a novel grouping method called Credit-driven adaptive Grouping (CreateG). By monitoring agents' credit—a metric that characterises an individual's contribution to the group—in our experiments, we found that low-credit agents within a group are often redundant or even detrimental to their peers due to the constraints of parameter sharing. For example, in the MMM2 scenario of SMAC, if a Medivac and a Marauder are assigned to the same group (sharing a single policy), the model likely converges towards a strategy optimal for the Marauder, such as frontline tanking. This is inherently suboptimal for a Medivac, which requires flanking for healing, leading to diminished rewards and, consequently, low-credit. Crucially, because the Medivac and Marauder receive vastly different rewards despite having similar observations and actions, the Medivac's trajectories introduce noise and conflicting gradients into the shared network, which misleads the Marauder and hinders the discovery of an optimal collaborative policy. Hence, the motivation of CreateG lies in achieving self-organising group structures through the iterative elimination of redundant (low-credit) members from existing groups. To realize this, from a dynamic perspective, we divide the entire training process into multiple phases, and at the end of each phase, CreateG only exclusively extracts low-credit individuals from their original groups and reallocates them to new groups. Thus, our method can adaptively divide groups according to scenario tasks and reasonably allocate the composition and quantity of members in each group. Furthermore, we design a hierarchical hypernetwork architecture tailored to this adaptive grouping mechanism.

We validate CreateG on three commonly used multi-agent benchmarks: SMAC tasks (hard and superhard scenarios), Google Research Football (GRF) [27] scenarios and Predator-Prey tasks [28]. Experimental results demonstrate that CreateG achieves state-of-the-art performance. Our ablation study provides experimental evidence for the effectiveness of each component of our method. The contributions of this paper are summarised as follows:

1. We propose CreateG, a novel adaptive grouping paradigm for cooperative MARL, which is designed to enhance policy heterogeneity in MARL via grouping. By labelling the credit of each agent in our experiments, we find that low-credit agents within a group are often redundant in cooperative tasks. In light of this, we design a credit-driven adaptive grouping mechanism that incrementally allocates redundant agents from each group to new groups in a phased manner, thereby forming multiple groups with heterogeneous policies that adapt to task requirements. Distinct from existing grouping methods, CreateG obviates the need for predefining the number of groups or the types/quantities of group members.
2. We conduct comprehensive validation of CreateG on three representative MARL benchmarks (SMAC hard/superhard scenarios, Google Research Football and Predator-Prey tasks), where it achieves state-of-the-art performance. Additionally, we perform ablation studies to provide empirical evidence for the effectiveness of each core component of our method.

2 | Related Work

CreateG aims to address the issue of lack of policy heterogeneity caused by the parameter sharing mechanism in MARL. There are mainly two types of approaches to enhance policy heterogeneity, namely at the identity level and the group level. Therefore, the following will briefly summarise the related work in two domains: identity-level diversity, group-level diversity.

2.1 | Identity-Level Diversity

Some studies tend to enhance the diversity of each agent. EOI [16] uses an intrinsic reward to encourage inter-agent division of labour and two regularisation terms to widen initial policy gaps. MAVEN [17] employs a hierarchical structure to control latent variables, deriving inter-agent mutual information from trajectories and latent data; maximising this boosts strategy divergence for more diverse actions. CDS [12] decomposes the state-action value function into shared and individual components, enabling partial experience sharing with agent-specific adaptations. CIA [18] suffers from insufficient individual diversity due to ambiguous credit assignment and increases diversity by minimising credit mutual information. However, in real-world scenarios, it is not always better for the strategies of individual agents to be more dispersed. In many cases, it is necessary to carry out grouped collaboration to accomplish tasks that cannot be well completed by the strength of individual agents alone.

2.2 | Group-Level Diversity

Grouping has long been recognized as an effective approach to enabling efficient coordination in multi-agent systems [29–32]. In MARL, grouping methods enhance agents' behavioural diversity through group-wise parameter sharing—under this definition, both role-based approaches and heterogeneous value decomposition methods can be categorised as grouping methods. RODE: Groups agents via action-effect clustering to

form role-specific action spaces, with hierarchical role assignment updated at fixed temporal intervals based on local observations. RODE Tonghan [23] proposes a hierarchical framework that clusters actions into role-specific spaces based on environmental effects. In contrast, ROMA [33] learns emergent latent role embeddings from observation-conditioned Gaussian distributions. For dynamic and context-aware grouping, ACORM [24] applies k-means clustering to agent trajectory embeddings and then derives role embeddings via contrastive learning. SMART lan clusters learned action embeddings into role prototypes and assigns roles sequentially via a causal-attention policy. VAST [25] demonstrates that flat value function factorization (VFF) schemes lead to performance bottlenecks and explores value factorization for sub-teams with predetermined group numbers. GoMARL [21] introduces a "select-and-kick-out" scheme for automatic grouping. Graph-based and hybrid strategies further enrich grouping paradigms: GACG [20] utilises graph attention to model pairwise agent interactions, optimising sub-team structures for cooperative performance. QTypeMix [34] enhances value decomposition by grouping agents into homogeneous or heterogeneous sub-teams, aligning group structures with specialised cooperative strategies. HYGMA [35] combines pre-defined task partitions with dynamic role adaptation, striking a balance between computational efficiency and environmental responsiveness. However, most existing grouping approaches require predefining key hyperparameters such as the number of groups and the composition and size of each group's members. These hyperparameters must be configured individually for different scenarios, and their optimal setup is challenging without adequate expert knowledge.

Our method, CreateG, aims to achieve group-level diversity. Compared with the aforementioned grouping methods, it designs a novel adaptive grouping mechanism, which eliminates the need to predefine the number of groups or the composition of intra-group members, and achieves adaptive grouping via a strategy of multi-round dynamic filtering, thereby allowing better adaptation to different scenarios without requiring prior domain knowledge.

3 | Preliminaries

3.1 | Dec-POMDP

We examine a fully cooperative multi-agent scenario. A Dec-POMDP is defined as a tuple $\langle \mathcal{A}, \mathcal{S}, \mathcal{U}, P, r, \Omega, \mathcal{O}, \gamma \rangle$ [36]. Here, $\mathcal{A} = \{a_k\}_{k=1}^{K}$ denotes a collection of K agents. $\mathcal{S}$ stands for the global state space of the environment. At every time step t, each agent $a_k \in \mathcal{A}$ gets a personal partial observation $o_t^k \in \Omega$, which is generated by the observation function $\mathcal{O}(s_t, a_k)$. After that, each agent selects a corresponding action $u_t^k \in \mathcal{U}$, thus forming a joint action $\mathbf{u_t} \in \mathcal{U}^K$. This joint action leads to a transition to the subsequent state s_{t+1} based on the state transition function $P(s_{t+1}|s_t, \boldsymbol{u_t}) : \mathcal{S} \times \mathcal{U}^K \times \mathcal{S} \to [0, 1]$. All agents utilise the same reward function $r(s_t, \boldsymbol{u_t}) : \mathcal{S} \times \mathcal{U}^K \to \mathbb{R}$, and $\gamma \in (0, 1)$ is the discount factor. Each agent a_k possesses an action-observation history $\tau^k \in \mathcal{T} \equiv (\Omega \times \mathcal{U})^*$ and learns its own policy $\pi^k(u^k|\tau^k) : \mathcal{T} \times \mathcal{U} \to [0, 1]$, with the aim of jointly maximising the discounted return $R_t = \sum_{i=0}^{\infty} \gamma^i r(s_{t+i}, \boldsymbol{u_{t+i}})$. The joint action-observation history is defined as $\tau \in \mathcal{T}^K$. The joint policy π gives rise to a joint value function $Q_{tot}^t(s_t, \mathbf{u_t}) = \mathbb{E}_{s_{t+1:\infty}, \boldsymbol{u_{t+1:\infty}}}[R_t|s_t, \boldsymbol{u_t}, \pi]$, which represents the expected discounted return under the specified policy.

3.2 | The CTDE Framework

The paradigm of centralised training with decentralised execution (CTDE) garners attention in cooperative MARL for effective policy learning [7]. Agents learn decentralised policies from local observations at execution, yet access global info in centralised training. One of the promising ways to exploit the CTDE framework is VD, where agents learn individual utilities by optimising the joint action-value function. A mixing network with params θ_v approximates Q_{tot}, merging individual action values into $Q_{tot} = f(q; \theta_v)$ ($q = \left[Q^k\right]_{k=1}^{K}$, Q^k for agent a_k with θ_π). Learnable $\theta = \{\theta_\pi, \theta_v\}$ update via minimising:

$$\mathcal{L}_{TD}(\theta) = \mathbb{E}_{\mathcal{D}}\left[\left(y_t - Q_{tot}(s_t, \mathbf{u_t}; \theta)^2\right)\right], \tag{1}$$

where $\mathbb{E}[\cdot]$ is expectation, $\mathcal{D}$ is replay buffer, $y_t = r_t + \gamma \cdot \max_{\mathbf{u_{t+1}}} Q(s_{t+1}, \mathbf{u_{t+1}}; \theta^-)$ as TD target.

Table 1 provides a summary of the main notations and their corresponding definitions used throughout this paper for the reader's convenience.

4 | Method

In this section, the overview of CreateG (see: Section 4.1) is first described, then we present the core method details of CreateG (see: Section 4.2). Finally, some performance analyses are given (see: Section 4.3).

4.1 | Overview of CreateG

Before formally introducing our method, we formalise the core concept of "group" in CreateG: A group is defined as a set of

TABLE 1 | Main notations and definitions.

Notation	Description
$\mathcal{A}$	Set of all agents in the system
$\mathcal{G}$	Set of groups formed by agent partitioning
G_j	The j-th group in $\mathcal{G}$
a_i	The i-th agent in $\mathcal{A}$
Cre_{ji}^t	Credit value of agent a_i in group G_j at time step t
$\mathcal{C}_j$	Credit vector of group G_j: $\mathcal{C}_j = \left[Cre_{ji}^t\right]_{a_i \in G_j}$
Q_{G_j}	Group-level Q-value for group G_j
$Q_G^{\text{tot}}(s_t, \mathbf{u}_t)$	Global return for all groups at state s_t with actions $\mathbf{u}_t$
$\alpha_{ji}(t)$	Time-varying importance weight of group sensitivity loss $\mathcal{L}_g(\theta)$

agents sharing policy network parameters, which is formally formulated as follows:

Definition 1. (Group). *Let $\mathcal{A}$ denote the set of agents and $\mathcal{P}$ denote the set of policy networks. Define a mapping $f : \mathcal{A} \to \mathcal{P}$. $\forall p \in \mathcal{P}$, we define a group $G_p = \{a \in \mathcal{A} : f(a) = p\}$. The following properties hold:*

1. $\forall p_1, p_2 \in \mathcal{P}, p_1 \neq p_2 \Rightarrow G_{p_1} \cap G_{p_2} = \varnothing$, which means agents from different groups are disjoint.
2. $\bigcup_{p \in \mathcal{P}} G_p = \mathcal{A}$, indicating that the union of all groups constitutes the entire set of agents.

The "credit" utilised in CreateG adheres to the mainstream definition in MARL [10, 11], referring to the contribution ratio of the individual agent's Q-value (Q_i) to the total Q-value (Q_{tot}). We adopt it to evaluate the contribution of agent a_i relative to group G_j, which is formulated as follows:

$$Cre_{ji}^{t} = \frac{\partial Q_{G_j}}{\partial Q_i}. \tag{2}$$

The overall architecture of CreateG, illustrated in Figure 1, follows a dynamic and iterative process. Initially, all agents are aggregated into a single group. As training progresses, the grouping configuration $\mathcal{G}$ evolves continuously. CreateG computes the agent credit Cre_{ji}^{t}, quantifying each agent's contribution or performance within its group every c time steps during training. The system re-evaluates and re-groups agents every c time steps, ensuring continuous optimization. This iterative cycle persists until the grouping configuration stabilises. Our method leverages a composite loss function comprising three key components: $\mathcal{L}_g$ for accelerating convergence, $\mathcal{L}_d$ for enforcing inter-group policy diversity, and $\mathcal{L}_{\mathcal{TD}}$ (proposed by Double Q-learning [37]) to enhance overall performance. These components synergistically optimise training efficiency and effectiveness.

4.2 | Core Method Details of CreateG

First, we propose a grouping strategy named the Credit-driven adaptive grouping mechanism, which is designed to learn a mapping, denoted as $f_g : \mathcal{A} \mapsto \mathcal{G}$, which partitions agents $\mathcal{A} = \{a_1, a_2, ..., a_N\}$ into groups $\mathcal{G} = \{G_1, G_2, ..., G_M\}$ with the ultimate goal of maximising the global return $Q_G^{tot}(s_t, \mathbf{u}_t)$.

The Credit-driven Adaptive Grouping Mechanism first divides the entire training process into several intervals with a fixed time step c. To ensure stable credit estimation for each training phase, c is set to a relatively large value (default: 100,000). At the end of each training phase, we calculate the credit value for each agent; the agents moved to the new group are those with:

$$\text{Cre}_{ji}^{t} < \Delta, \quad \text{where} \quad \Delta = \left(\frac{1}{|G_j|} \sum_{i \in G_j} \text{Cre}_{ji}^{t} \right) \cdot \delta, \quad 0 < \delta < 1. \tag{3}$$

To adapt to the credit-driven adaptive grouping mechanism, we design a hierarchical hypernetwork architecture. As illustrated in the Figure 2, the detailed structure of a single group is built upon the QMIX [10] framework, with an additional group mixing network θ^G integrated to estimate the group Q-value Q_{G_j}, which is formulated as:

$$Q_{G_j} = W_2 \cdot f_{elu}\left(W_1 \cdot \mathcal{Q}_I^T + b_1\right) + b_2, \tag{4}$$

where W_1, W_2 are weight matrices, b_1, b_2 are bias vectors, and f_{elu} denotes the exponential linear unit activation function. This architecture allows the model to capture complex cooperative strategies emergent from agent interactions.

Subsequently, $\frac{\partial Q_{G_j}}{\partial Q_i}$ is calculated as follows:

$$dQ_{G_j} = W_2 F W_1 d\mathcal{Q}_I, \tag{5}$$

where F is a diagonal matrix and $F_{ii} = \frac{df_{elu}(x_i)}{x_i}|_{x_i = (W_1 Q + b_1)_i}$.

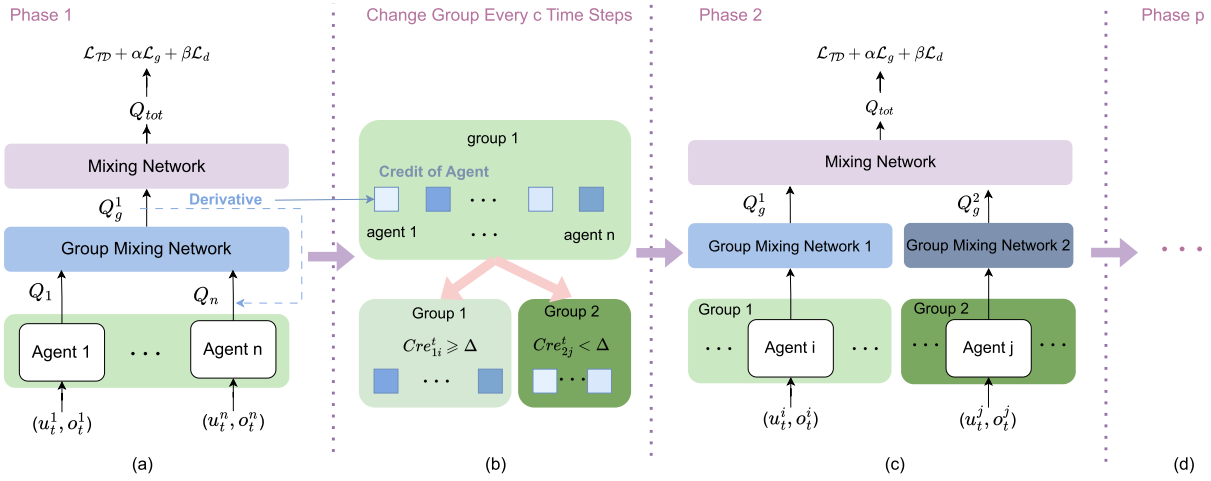


FIGURE 1 | The overall architecture of CreateG. (a) Initial state: all agents are grouped into one single group; (b) Credit-based grouping mechanism: agents are re-grouped according to their credit values; (c) Algorithmic structure after grouping: illustrates the operational framework of the grouped agents; (d) The above process is repeated iteratively until a stable grouping is achieved.

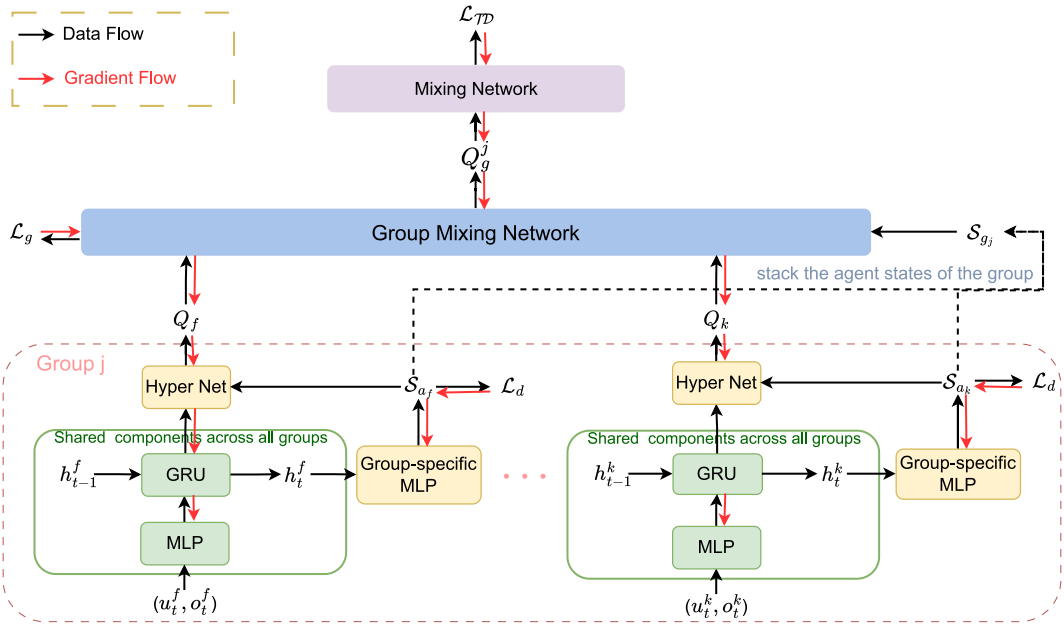


FIGURE 2 | Details of the hierarchical hypernetwork architecture: the new layer group mixing network is responsible for estimating the value of Q_{G_j}; the backbone of the agent net maintains the parameter-sharing, and only its decision head is replaced with a hypernetwork. For different groups, only the parameters of the group-specific MLP need to be adjusted. This design ensures policy diversity while effectively preserving the feature of experience sharing.

Another MLP θ_a is introduced to estimate agent state $\mathcal{S}_{a_i}$ based on its historical trajectory, feeding it as inputs to the hypernetwork. The group state $\mathcal{S}_{g_j}$ formed by stacking the agent states is fed to group mixing network θ^G.

After adding a new group, θ^{shared} is shared across all groups. The new group only updates the parameters of θ_a, thereby adjusting the agent policy network of different groups by modifying the input of the hypernetwork. This design enables agents to achieve policy diversity after grouping while maximising the retention of shared experiences.

The initial parameters of θ_a and the group mixing network θ^G are initialised through a momentum update from the parameters of the previous group, followed by independent training for the new group. This parameter initialisation process is referred to as "Warm-Start Parameter Initialisation across Groups" which is formulated as:

$$\theta_{G_{new}} = \frac{1}{|\mathcal{G}| - 1} \sum_{\substack{G' \in \mathcal{G} \\ G' \neq G_{new}}} \theta_{G'}, \tag{6}$$

where $\mathcal{G}$ represents the set of formed groups.

To accelerate the stabilisation of grouping in the later stage of training, we define a group sensitivity loss:

$$\mathcal{L}_g(\theta) = \sum_{j=1}^{M} \sum_{i=1}^{|G_j|} \alpha_{ji}(t) \cdot \left\| Cre_{ji}^t \right\|_1, \tag{7}$$

where $\alpha_{ji}(t)$ is a time-varying importance weight defined as $\alpha_{ji}(t) = \mu \cdot (1/e)^{t/c}$. Here, t represents the current training time step, c denotes the group reorganization interval (consistent with the interval used for $Cret_{ji}$ re-evaluation in our previous response), and μ is a preset constant to scale the weight magnitude, and $\|\cdot\|_1$ denotes the l_1-norm penalty. This loss term encourages the model to assign consistent credit values over time, reducing unnecessary group reorganizations and fostering stable policy development.

To promote more diverse policies in the newly trained policy networks, we introduce a group diversity loss to maximise the distance between the inputs of the hypernetworks:

$$\mathcal{L}_d(\theta) = \sum_{j=1}^{M} \sum_{a_f \in G_j} \left(\sum_{a_h \in G_j, a_h \neq a_f} \cos\left(\mathcal{S}_{a_h}, \mathcal{S}_{a_f}\right) - \sum_{a_k \notin G_j} \cos\left(\mathcal{S}_{a_k}, \mathcal{S}_{a_f}\right) \right). \tag{8}$$

The training process focuses on minimising a composite loss function, which combines a temporal-difference (TD) loss (Equation 1) term with a novel group divergence loss component. This integrated loss structure facilitates end-to-end optimization for policy learning.

$$\mathcal{L}(\theta) = \mathcal{L}_{TD}(\theta) + \alpha \mathcal{L}_g(\theta) + \beta \mathcal{L}_d(\theta), \tag{9}$$

where θ includes all parameters in the model, α, β are the regularisation coefficients for the group sensitivity loss and the group diversity loss, respectively

Algorithm 1 demonstrates the overall training process of CreateG.

ALGORITHM 1 | Overall training process of CreateG.

Input: Agents set $\mathcal{A}$, total steps T, grouping interval c, grouping threshold Δ
Output: Learned groups $\mathcal{G} = \{G_1, G_2, ..., G_M\}$
Initialize all agents in one group: $\mathcal{G} = \{G_1\} = \{\mathcal{A}\}$
Randomly initialize agent networks $\Theta^a = \{\theta_1^a, ..., \theta_n^a\}$, group mixing networks $\Theta^G = \{\theta_1^G\}$ and mixing network θ^{mix}
$t \leftarrow 0$
while $t < T$ **do**
  $t \leftarrow t + 1$
  Train $\Theta^a; \Theta^G; \theta^{mix}$ using the loss function proposed in Equation (9)
  if $t \bmod c = 0$ **then**
    for each group $G_j \in \mathcal{G}$ **do**
      Calculate agent credits within G_j using Equation () (2)
      $G_j \leftarrow \{a_i \in G_j | credit[i] \geq \Delta\}$
      if $\{a_i \in G_j | credit[i] < \Delta\}$ is not empty **then**
        Create a new group G_{new}
        $G_{new} \leftarrow \{a_i \in G_j | credit[i] < \Delta\}$
        $\mathcal{G} \leftarrow G_{new} \cup \mathcal{G}$
        Create a new group mixing network θ_{new}^G for this G_{new}
        $\Theta^G \leftarrow \theta_{new}^G \cup \Theta^G$
        initialize the agent network of all member of G_{new} through Equation (6)
      end if
    end for
  end if
end while

4.3 | Performance Analysis

This section presents a theoretical performance analysis of the proposed CreateG, focussing on the computational complexity. We analyse how the algorithm scales with respect to the number of agents N, total training steps T, grouping interval c, and model parameter dimensionality d. These results provide insights into the efficiency and scalability of the method. The main computational cost of CreateG comes from two major components:

1. Training Phase: At each training step, all groups are trained independently. Let M_t denote the number of groups at time step t, and let n_i be the number of agents in group i. Assuming that the cost of training a single agent is proportional to the parameter dimension d, the cost of training a group of size n_i can be expressed as $O(n_i \cdot d)$. Therefore, the total cost for one training iteration is $\sum_{i=1}^{M_t} O(n_i \cdot d) = O(N \cdot d)$

Since this is repeated over T training steps, the total training cost within one phase becomes $O(T \cdot N \cdot d)$

2. Group Reorganization: Every c steps, the algorithm re-evaluates agent credits and reorganises them into new groups. This involves Evaluating credit values: $O(N)$ and Copying parameters for newly formed groups: $O(M \cdot d)$

Given that this operation occurs T/c times, the total cost of re-grouping is $O\left(\frac{T}{c} \cdot (N + M \cdot d)\right)$

Combining both components, the overall time complexity of the algorithm is $O\left(T \cdot N \cdot d + \frac{T}{c} \cdot (N + M \cdot d)\right)$

Remark 4.1. The algorithm scales linearly with the number of agents N and training steps T. A larger grouping interval c reduces the frequency of re-grouping and thus lowers overhead. However, increasing the maximum number of groups M increases the computational burden due to additional parameter copying.

5 | Experiment

In this section, we design experiments to answer the following questions: (1) Can the adaptive grouping mechanism of CreateG improve the overall performance? (2) Can CreateG assign group members adaptively to maximise effectiveness? (3) Is WSPI-G more effective compared to training from Random initialisation? (4) What impact does the setting of the grouping threshold have on the performance of CreateG? (5) How does the setting of the grouping gap affect the final result?

5.1 | Experimental Setup

We compare all the methods in six Hard and Super Hard StarCraft II micromanagement tasks (SMAC) [38] and three challenging Google Research Football (GRF) [27] scenarios and two TAG [28] scenarios. To ensure fair comparisons,our experiments are based on the PyMARL2 [39] framework proposed to compare algorithms fairly. CreateG and all the baseline methods in our paper are trained with 8 parallel runners, which can facilitate faster convergence of methods. We use one NVIDIA 4090 GPU for training. Please refer to Table 2 for detailed hyperparameters.

5.2 | Performance Comparison

5.2.1 | *Benchmarks*

To evaluate the performance and scalability of our proposed method, we conducted extensive experiments across three diverse Multi-Agent Reinforcement Learning (MARL) benchmarks. These environments provide a variety of challenges, including high-dimensional state spaces, sparse rewards, and the need for complex coordination.

1. *StarCraft Multi-Agent Challenge (SMAC)* [38] is a widely recognized benchmark for cooperative MARL, focussing on micromanagement tasks within the StarCraft II

environment. In this study, we evaluated our approach on six maps classified as "Hard" and "Super Hard." These maps encompass a comprehensive range of scenarios, featuring both homogeneous and heterogeneous agent setups, as well as symmetric and asymmetric configurations. All the experiments are carried out with difficulty = 7.

2. *Google Research Football (GRF)* [27] is a physics-based simulator where agents must learn to coordinate in a high-stakes sports environment. We utilised three standard scenarios to test our model's ability to handle complex movements and offensive strategies: 3-vs-1-with-keeper, counterattack-easy, and pass-and-shoot-with-keeper. These maps require precise timing and spatial awareness to successfully bypass the opponent's defense and score.
3. *Target-Adversary Game (MPE-TAG)* [28] is a challenging task based on the Multi-Agent Particle Environment (MPE). In this scenario, a group of agents must cooperate to chase and capture several adversaries within a map containing randomly generated obstacles. While the agents receive a collective reward for colliding with an adversary, the adversaries are programmed to move faster, making individual pursuit ineffective. Consequently, the agents must develop sophisticated collaborative strategies to surround their targets. To evaluate the scalability of different methods, we tested both the standard configuration (10 agents vs. 3 adversaries) and an extended, high-density setup (20 agents vs. 5 adversaries).

5.2.2 | *Baselines*

We compare CreateG with the several state-of-art baseline algorithms. We categorise these baselines into two distinct types based on the core criterion of whether they adopt a grouping strategy. Below, we provide a brief introduction of each method we used:

5.2.2.1 | Non-Grouping Methods.

1. QMIX [10] & QPLEX [40]: These are representative value-decomposition approaches. QMIX employs a monotonic mixing network, while QPLEX utilises a duplex duelling architecture to maintain the Individual-Global-Max (IGM) principle. However, both suffer from inherent agent behaviour consistency issues, as the lack of group division limits their ability to represent specialised or diverse strategies.
2. CIA [18]: This method leverages contrastive learning to enhance credit-level discrimination among agents. While it improves the representation of individual identities, it still operates within a non-grouping framework.

5.2.2.2 | Grouping Methods. In this study, a "group" is defined as a collective of agents that share the same policy network. Under this definition, both role-based and heterogeneous value decomposition methods can be also categorised as grouping approaches. We compare our work against the following:

1. GoMARL [21] implements a dynamic grouping strategy through a "select-and-kick-out" mechanism, allowing the team to adaptively refine group memberships during the learning process.
2. RODE Tonghan [23] achieves grouping based on role decomposition. It factorises the complex action space into smaller role-action spaces, grouping agents that share similar responsibilities.
3. GACG [20] focuses on learning an underlying coordination structure. It typically groups agents by identifying latent dependencies and constructing a dynamic graph where edges represent high-level coordination needs between specific subgroups.
4. SOG [26] selects the most diverse agents as "conductors" based on behaviour models and forms teams led by these conductors.

5.2.3 | *Performance Comparison on SMAC*

We conducted performance tests on the SMAC benchmark [38], which encompasses a diverse collection of maps populated with different quantities of agents. We conducted tests on our algorithm across six hard and super hard maps from the benchmark. The details of these maps' information are illustrated as Table 3.

TABLE 2 | Hyperparameter for CreateG.

Parameter	Value
Buffer size	5000
Batch size	128
Parallel batch size run	8
Hypernet embedding dimension	32
Change group interval c	100,000
Grouping threshold δ	0.7
TD lambda rate	0.6
Discount factor γ	0.99
Target network update interval T_{int}	200
Nonlinear activation function	ELU
Optimiser	Adam
Adam weight range	(0.9, 0.999)

TABLE 3 | SMAC task information.

Difficulty	Map	Ally agents	Enemy agents
Hard	1c3s5z	1 Colossi 3 Stalkers 5 Zealots	1 Colossi 3 Stalkers 5 Zealots
	8m-vs-9m	8 Marines	9 Marines
	10m-vs-11m	10 Marines	11 Marines
	2c-vs-64zg	2 Colossi	64 Zerglings
Super hard	Corridor	6 Zealots	24 Zerglings
	MMM2	1 Medivac 2 Marauders 7 Marines	1 Medivac 2 Marauders 8 Marines

As shown in the Figure 3, we tested six hard and super hard scenarios in SMAC. Compared with baseline algorithms, our algorithm has achieved promising results. It can be seen that the more complex the scenario, the better the effect of CreateG. This is because in complex scenarios, agents must achieve more diverse collaboration through correct grouping. Table 4 further intuitively demonstrates that CreateG outperforms the baseline in most scenarios, both in terms of performance and stability.

As shown in an example from the corridor scenario in Figure 4, CreateG demonstrates how the learned grouping mechanism works. CreateG divides agents into two groups: $\{\{a1\}, \{a2, a3, a4, a5, a6\}\}$. Here, agent $a1$ is tasked with attracting the majority of enemies and kiting them, buying time for the other five agents to eliminate the remaining smaller enemy force. Notably, when $a1$ is on the verge of being defeated, all agents in Group 2 remain alive with healthy HP, and are close to wiping out their confronting enemies. This allows the five agents to retain sufficient combat effectiveness to tackle the remaining enemies in subsequent battles.

We further analyse a detailed visualisation of the grouping results in another "Super Hard" SMAC scenario: 27m-vs-30m.In this scenario involving 27 agents, CreateG successfully learns to partition the team into three distinct functional subgroups: Group 1 $\{a_1, ..., a_{12}\}$: This group maneuvers towards the upper

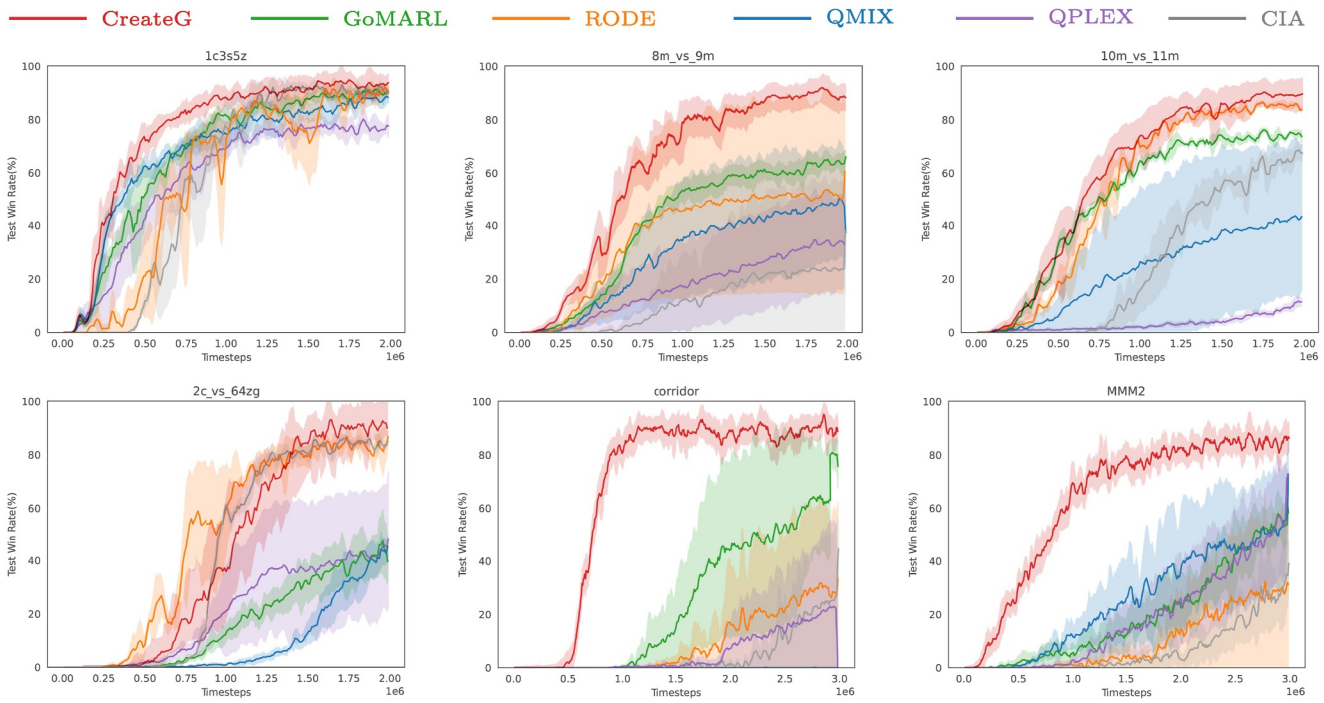


FIGURE 3 | Performance of CreateG and baselines on six hard and superhard maps of the SMAC. The *x*-axis represents the time steps (in millions), while the *y*-axis quantifies the test win rate in the games.

TABLE 4 | Performance comparison of CreateG and baselines on SMAC maps. Values represent mean ± standard deviation over 5 random seeds.

Map	Steps	CreateG	GoMARL	RODE	QMIX	QPLEX	CIA
Corridor	1e6	**81.0 ± 4.2**	2.1 ± 1.5	0.5 ± 0.2	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	3e6	**90.2 ± 3.8**	80.5 ± 9.2	35.2 ± 12.5	28.0 ± 8.1	25.0 ± 7.2	20.3 ± 7.4
MMM2	1e6	**55.2 ± 6.1**	5.3 ± 2.4	10.1 ± 3.8	2.5 ± 1.1	1.2 ± 0.5	8.4 ± 3.2
	3e6	**88.3 ± 4.5**	58.1 ± 10.3	32.4 ± 14.2	55.0 ± 11.5	50.2 ± 9.4	38.5 ± 8.1
1c3s5z	1e6	**82.1 ± 3.5**	80.5 ± 5.2	62.0 ± 8.4	78.2 ± 4.1	65.4 ± 6.2	80.1 ± 3.9
	2e6	**91.5 ± 2.8**	88.2 ± 4.3	85.0 ± 7.1	88.0 ± 3.5	78.3 ± 5.1	89.2 ± 3.2
8m_vs_9m	1e6	**75.4 ± 5.4**	50.2 ± 10.8	38.1 ± 17.5	40.5 ± 6.1	22.1 ± 4.5	15.3 ± 7.2
	2e6	**89.1 ± 4.1**	65.4 ± 8.2	55.2 ± 19.4	50.1 ± 7.5	35.4 ± 6.8	22.0 ± 9.1
10m_vs_11m	1e6	**78.2 ± 4.8**	68.3 ± 5.5	75.1 ± 6.2	32.4 ± 15.8	5.1 ± 1.8	15.0 ± 8.4
	2e6	**90.3 ± 3.2**	78.4 ± 6.1	88.2 ± 5.4	42.1 ± 7.2	12.3 ± 3.5	68.2 ± 5.9
2c_vs_64zg	1e6	38.5 ± 7.2	15.3 ± 4.8	**55.0 ± 15.2**	5.2 ± 2.1	35.1 ± 8.5	39.4 ± 6.4
	2e6	**92.1 ± 3.5**	40.2 ± 11.4	85.4 ± 6.8	39.3 ± 8.2	45.0 ± 10.1	82.2 ± 9.7

Note: The bold values indicate the best performance among all compared methods.

flank of the map; Group 2 $\{a_{13}, ..., a_{18}\}$: This central group maintains its position to draw enemy fire and hold the line; Group 3 $\{a_{19}, ..., a_{27}\}$: This group maneuvers towards the lower flank. As shown in the newly added Figure 5, these three groups collaborate to execute a surrounding maneuver against the 30 enemy units. This result clearly indicates that our method can dynamically identify and form more than two groups based on the tactical requirements of the environment, leading to more effective coordination in large-scale battles.

5.2.4 | *Performance Comparison on GRF*

We further evaluated the performance of CreateG on three challenging GRF offensive scenarios with three baselines (GoMARL, QMIX, QPLEX). Agents in GRF must precisely coordinate their timing and positions to mount effective offensive policies, capitalising on the narrow windows of opportunity available. Since rewards are solely based on scoring goals, the tasks in GRF are inherently more complex than those in SMAC battles. As such, the comparison conducted within the GRF framework serves as an additional validation of the efficacy of our proposed method.

As is shown in Figure 6 and Figure 7, performance tests are conducted on three academy scenarios of Google football tasks as follows:

1. *3-vs-1-with-keeper*: Here, three of our players work together inside the penalty box in an attempt to score. They are up against one defending player and the goalkeeper. Two of our players take positions on either side of the penalty area, while the third stands in the middle of the field to take the kickoff.
2. *counterattack-easy*: This situation is comparable to Counterattack-Hard, but with one fewer defender present, making it slightly less challenging.
3. *pass-and-shoot-with-keeper*: A single offensive agent (player) navigates the penalty area against a goalkeeper, aiming to score via precise passing and shooting. The agent must dribble, pass, and time shots while the keeper dynamically blocks attempts.

Against the backdrop of intense competition in multi-agent reinforcement learning methods, our proposed algorithm, CreateG, stands out with its innovative architecture that integrates advanced neural network modules and novel reward shaping policies. Rigorous evaluations across a diverse array of scenarios, including complex multi-agent coordination tasks and dynamic adversarial environments, have been conducted to comprehensively assess CreateG's efficacy. Experimental results reveal that CreateG consistently exhibits highly competitive performance, outshining many state-of-the-art algorithms. Notably, it is the sole method capable of surpassing a 60% score reward threshold across all tested scenarios. This achievement underscores CreateG's robustness and adaptability in handling varying task complexities and environmental dynamics. In the second testbed, which features intricate agent interactions and stringent performance requirements, CreateG demonstrates remarkable efficiency. It achieves optimal solutions with fewer training iterations compared to counterparts, reducing the overall computational cost by approximately 40%. Such outstanding performance in

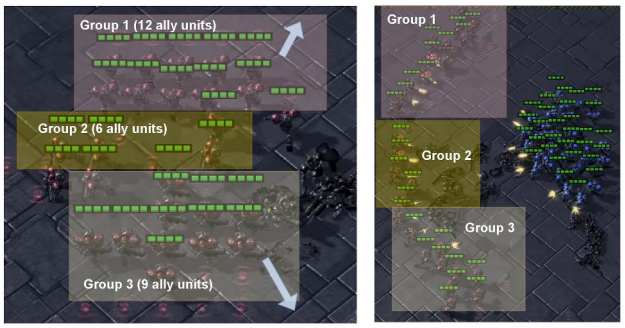


FIGURE 5 | The learnt groups $\{\{a1, ..., a12\}, \{a13, ..., a18\}, \{a19, ..., a27\}\}$ is explicable to the 27m-vs-30m (super hard) scenario. (after 3 million training steps).

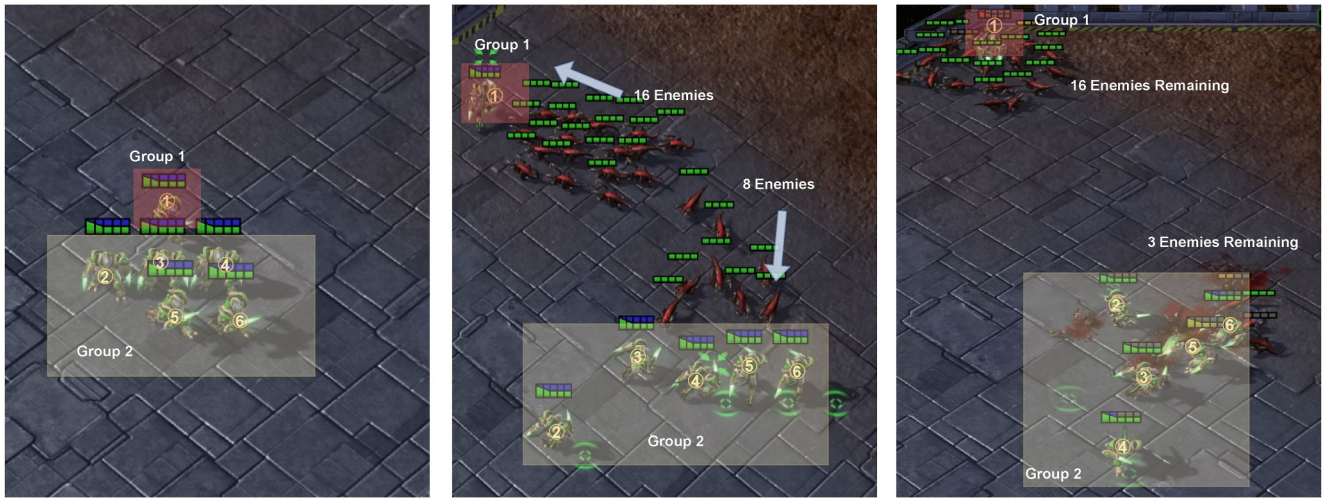


FIGURE 4 | The learnt groups $\{\{a1\}, \{a2, a3, a4, a5, a6\}\}$ is explicable to the corridor combat situation. (after 2 million training steps).

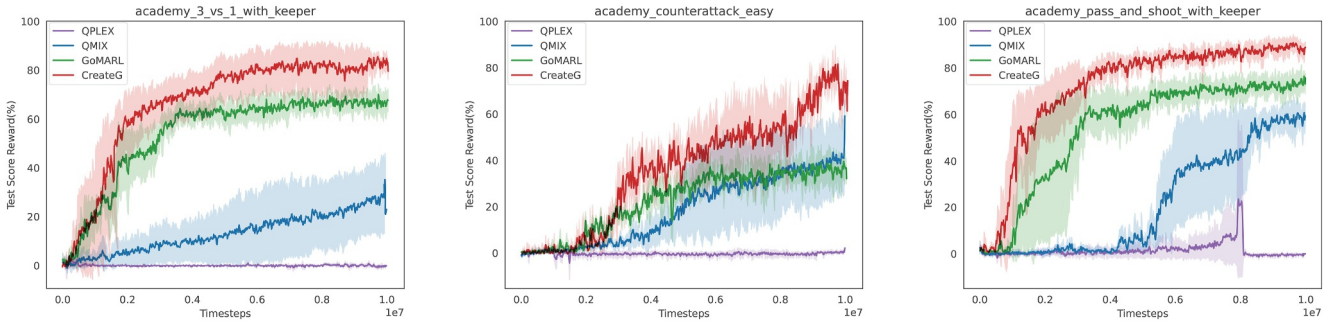


FIGURE 6 | Learning curves of CreateG and comparative algorithms on three Google football tasks (averaged results of five random seeds).

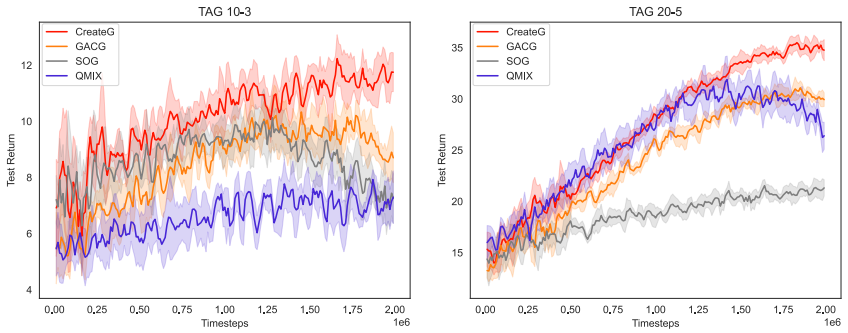
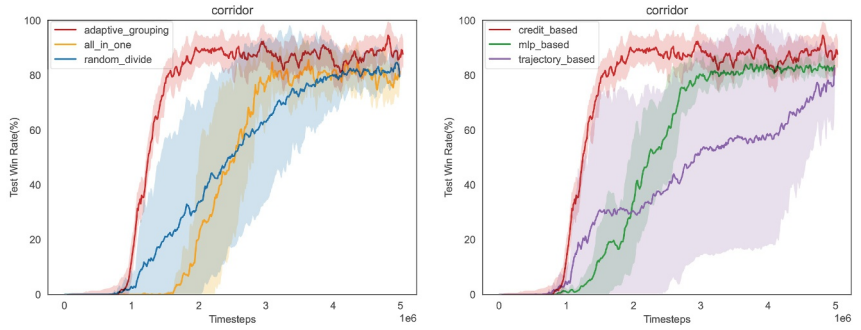


FIGURE 7 | Ablation study on credit-driven adaptive grouping mechanism. The curves represent the average results of five random seeds.



(a) Validation of the number of groups generated by CreateG's grouping mechanism.

(b) Validation of the member composition effectiveness in groups generated by CreateG's grouping mechanism.

FIGURE 8 | Learning curves of CreateG and comparative algorithms on two large-scale TAG tasks (averaged results of five random seeds).

this challenging testbed further validates the exceptional transferability of CreateG, indicating its potential to be effectively applied to a wide spectrum of multi-agent systems beyond the tested scenarios.

5.2.5 | *Performance Comparison on TAG*

We conducted a series of comparative experiments in the TAG benchmark. We compared CreateG against three baseline algorithms: GACG, SOG, and QMIX, across two different task scales: TAG 10-3 and TAG 20-5.

As illustrated in the training curves in Figure 8, CreateG consistently outperforms all baseline methods in terms of the average test return throughout the 2-million timestep training process. In both the 10-3 and 20-5 configurations, CreateG exhibits a superior learning rate and achieves the highest final mean return, reaching approximately 12 in the smaller environment and over 35 in the larger scenario.

While baseline methods like GACG and QMIX show competitive growth in the early stages, their performance tends to saturate or even decline as training progresses—a trend particularly visible in the TAG 20-5 results after 1.5 million timesteps. In contrast, CreateG maintains a more stable and robust upward trajectory. The SOG method, while competitive in the TAG 10-3 setting, fails to scale effectively to the TAG 20-5 environment, resulting in a significant performance gap compared to CreateG. These results demonstrate that CreateG possesses superior scalability and effectiveness in handling the complex coordination and adversarial dynamics inherent in the TAG environment.

5.3 | Ablation Study

5.3.1 | *Effectiveness of the Grouping Mechanism*

In this section, we delve into the question of whether the adaptive grouping mechanism embedded in CreateG can enhance the overall effectiveness of the algorithm, focussing on two key dimensions: the rationality of the number of groups and the optimality of group member composition. These two aspects are critical for evaluating the mechanism's ability to dynamically adjust grouping strategies in response to environmental changes and task requirements.

To ensure the validity and generalisability of our findings, we selected the "corridor" map from the SMAC benchmark as the testing environment. This map is characterised by its intricate terrain layout and complex agent interaction patterns, where suboptimal grouping can lead to significant performance degradation—making it an ideal setting to isolate and assess the impact of grouping strategies. Notably, to eliminate confounding variables and solely focus on the influence of the grouping mechanism itself, all comparative methods in this study adopt the identical underlying architecture as CreateG, with only the grouping strategies being modified.

To investigate how the number of groups affects algorithm performance, we designed three sets of comparative experiments:

1. Single-group baseline: All agents are aggregated into a single group, where decisions are made collectively without sub-group division.
2. Agents are randomly assigned to a varying number of groups (ranging from 2 to 5, consistent with the possible group counts generated by CreateG). This experiment was repeated 10 times with different random seeds to account for stochasticity in group assignments.
3. CreateG's adaptive grouping: The number of groups is dynamically determined by CreateG's built-in mechanism, which adjusts based on real-time agent states and task progress.

Next, we examined the impact of group member composition by fixing the number of groups to match that generated by CreateG (3 groups in the corridor scenario) and comparing three grouping criteria:

1. MLP-based allocation: A standard multi-layer perceptron (MLP) is used to assign agents to groups based on Q_i.
2. Trajectory-similarity grouping: Agents are clustered into groups based on the similarity of their historical movement trajectories, calculated using cosine similarity between trajectory vectors.
3. CreateG's adaptive composition: Group members are selected dynamically by CreateG, considering both local interaction patterns and global task objectives.

Collectively, as illustrated in 8, these results demonstrate that both the number of groups and the composition of group members, as determined by CreateG's adaptive mechanism, contribute significantly to its enhanced performance. This validates the mechanism's capacity to optimise grouping strategies in a context-aware manner, thereby reinforcing the algorithm's robustness and adaptability across complex multi-agent scenarios.

5.3.2 | *Influence of Warm-Start Parameter Initialisation Across Groups*

In this part, we conducted experiments to assess the impact of Warm-Start Parameter Initialisation across Groups (WSPI-G) on overall performance, using the corridor map as the experimental environment. As illustrated in Figure 9, the WSPI-G approach enhances overall effectiveness. Without model transfer, the policy network of the new group would theoretically have to start training from scratch. Consequently, it cannot inherit the effective experience accumulated from previous training, which is expected to degrade the overall performance. In this study, we aim to verify whether this hypothesis holds true.

5.4 | Hyperparameter Study

In this part, we tested the sensitivity of the algorithm to hyperparameters related to the grouping. The experiment was carried out by setting the grouping threshold δ to $\{0.6, 0.7, 0.8, 0.9\}$ and the grouping interval c to $\{50000, 100000, 200000, 500000\}$.

The results are illustrated in Figure 10. CreateG is relatively insensitive to grouping gaps. However, an extremely small grouping gap can compromise overall performance to some extent. A plausible explanation is that initiating the next grouping before the credit assignment has stabilised within a group may lead to incorrect groupings. Setting the parameter c to 100,000 proves sufficient to ensure the stabilisation of credit assignment within a group before the subsequent grouping operation. In contrast, CreateG is highly sensitive to the grouping threshold. When the grouping threshold is set too high, it can result in a higher rate of member attrition, thereby causing incorrect groupings.

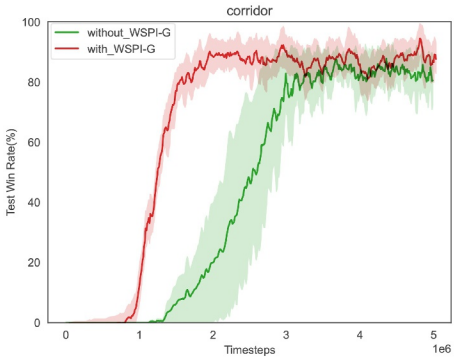
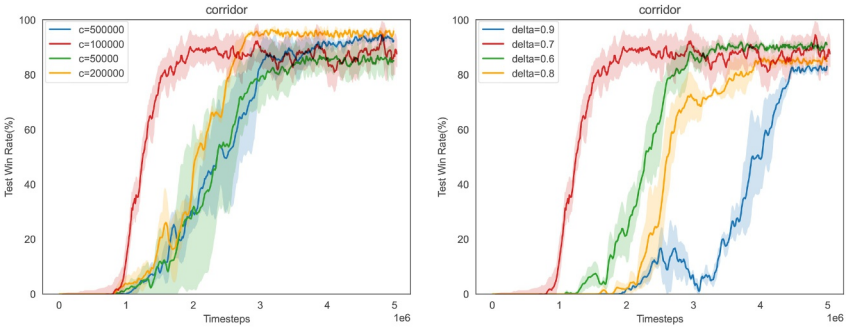


FIGURE 9 | Ablation study on the credit-driven adaptive grouping mechanism: compare with other grouping methods from the perspectives of group quantity and member composition.



(a) Influence of group interval.

(b) Influence of group threshold.

FIGURE 10 | Hyperparameter analysis of group-related parameters for the CreateG in the SMAC corridor map. The curves represent the average results of five random seeds.

In this section, we conducted a systematic sensitivity analysis to examine how the algorithm's performance responds to key hyperparameters associated with the grouping mechanism: the grouping threshold δ and the grouping interval c. These parameters play critical roles in regulating the dynamics of group formation: δ determines the similarity threshold for agent clustering, while c controls the frequency of grouping updates (measured in training steps). To quantify their impacts, we designed a full-factorial experiment using the corridor map from the SMAC benchmark, evaluating all combinations of $\delta \in \{0.6, 0.7, 0.8, 0.9\}$ and $c \in \{50,000, 100,000, 200,000, 500,000\}$. Each parameter setting was tested across 5 independent runs with different random seeds, and performance was assessed using average episode reward, success rate, and convergence speed. As visualised in Figure 10, the results reveal distinct sensitivity patterns to the two parameters:

5.4.1 | *Sensitivity to Grouping Interval c*

CreateG exhibits relative insensitivity to variations in the grouping interval c across most tested values. Performance remains stable when c is set to $\{50,000, 100,000, 200,000, 500,000\}$ steps, with average reward fluctuations within a 5% range. However, a notable performance degradation is observed when c is reduced to 50,000 steps. This aligns with our hypothesis that an excessively small grouping interval disrupts the stabilisation of credit assignment within groups. When group structures are updated before agents can fully learn cooperative behaviours and reward signals are properly propagated, the frequent reshuffling leads to inconsistent learning signals—hindering the formation of stable coordination patterns. Our findings confirm that setting $c = 100,000$ steps provides an optimal balance: it allows sufficient time for intra-group credit assignment to stabilise while maintaining adaptability to environmental changes.

5.4.2 | *Sensitivity to Grouping Threshold δ*

In contrast, CreateG demonstrates high sensitivity to the grouping threshold δ. Performance peaks at $\delta = 0.7$, with a success rate of 89%. As δ increases to 0.8 and 0.9, however, performance declines sharply: the success rate drops by almost 20% and 30% respectively, accompanied by increased variance across runs. This degradation stems from the overly strict similarity criteria imposed by high δ values, which trigger excessive member attrition from existing groups. Agents that marginally fall below the threshold are repeatedly reassigned, leading to fragmented group structures and disrupted cooperative strategies. Conversely, a lower threshold ($\delta = 0.6$) results in overly large and homogeneous groups, reducing the algorithm's ability to specialise in sub-tasks—though this effect is less pronounced than the penalties of excessively high δ.

These results highlight the importance of parameter calibration for the grouping mechanism. While the relative insensitivity to c simplifies practical deployment, the critical role of δ underscores the need for adaptive threshold adjustment in future iterations to enhance robustness across diverse scenarios.

6 | Discussion

This section presents the limitations, compute cost and future works of CreateG.

6.1 | Compute Cost

All experiments were conducted on hardware equipped with Intel(R) Xeon(R) Platinum 8474C CPUs and NVIDIA RTX 4090 GPUs (24 GB), with the software stack consisting of Python 3.8.10, PyTorch 1.11.1, PyYAML 6.0, numpy 1.21.5, and CUDA 11.3. We further report the detailed running time and GPU memory consumption of the proposed CreateG method and baseline approaches across two representative scenarios: the homogeneous setting ($10m_vs_11m$) and the heterogeneous setting (MMM2). Quantitative results in these tables (Tables 5 and 6) demonstrate that CreateG achieves a favourable trade-off between performance gains and computational costs. Specifically, while delivering superior grouping effectiveness for multi-agent systems, our method maintains acceptable levels of running time and GPU resource consumption. This efficiency highlights the practicality of CreateG, as it avoids the prohibitive computational overhead that often plagues advanced grouping mechanisms in MARL.

TABLE 5 | Running time and GPU consumption on $10m_vs_11m$.

Method	1M steps time (min)	GPU memory
QMIX	41.51 ± 4.24	3.11 GB
QPLEX	57.42 ± 8.33	4.42 GB
CIA	44.36 ± 6.69	3.98 GB
RODE	97.53 ± 10.72	10.33 GB
GoMARL	71.65 ± 9.21	5.13 GB
CreateG	60.03 ± 5.95	5.29 GB

TABLE 6 | Running time and GPU consumption on MMM2.

Method	1M steps time (min)	GPU memory
QMIX	75.83 ± 6.42	3.53 GB
QPLEX	72.59 ± 10.47	4.68 GB
CIA	67.38 ± 4.23	4.13 GB
RODE	132.73 ± 14.86	11.02 GB
GoMARL	60.24 ± 7.36	5.69 GB
CreateG	54.39 ± 5.28	6.14 GB

6.2 | Limitations and Future Work

Despite the effectiveness of CreateG in specific scenarios, it still has certain limitations that need to be addressed in future research. One of the primary limitations is the weak scenario generalisation ability. Since CreateG adopts the credit generated during training as the key basis for grouping, it is only applicable to fixed scenarios where the training environment and task settings remain unchanged. Once the scenario changes, the pre-trained credit model may no longer be applicable, leading to a significant decline in grouping performance. To tackle this limitation, future work will focus on enhancing the scenario transfer capability of the method. Specifically, we will explore more effective cross-scenario adaptation mechanisms that can reduce the dependence on scenario-specific training credit. Additionally, we plan to evaluate the improved method on the SMACv2 [41] benchmark, which incorporates sufficient randomness to challenge contemporary multi-agent reinforcement learning algorithms. This will provide a more comprehensive and rigorous validation of the scenario generalisation ability of the method.

Another limitation is the irreversibility of the grouping mechanism. Once an agent is assigned to a new group, it cannot return to its original group. This irreversibility may limit the flexibility of the grouping strategy, especially in dynamic scenarios where the optimal grouping structure may change over time. To address this issue, future research will consider introducing more grouping backtracking methods [42, 43]. These methods will enable the adjustment of grouping results based on real-time environmental changes and agent performance, allowing agents to be reallocated between groups when necessary, thereby improving the adaptability and flexibility of the grouping system.

7 | Conclusion

In this paper, we introduced Credit-driven Adaptive Grouping (CreateG), a novel approach that addresses the issue of policy diversity within MARL, which is constrained by the parameter-sharing mechanism employed in most value decomposition methods. Existing methods either incur substantial computational costs, lack interpretability, or exhibit high variance in results. CreateG utilises a credit-based adaptive grouping mechanism, derived from the Shapley value, to improve policy diversity while maintaining computational efficiency. Additionally, we propose a paradigm of policy transfer and independent training, enabling both shared policies within components and self-adaptation. Our extensive experimental evaluations demonstrate that CreateG consistently outperforms current leading methods. An ablation study further validates the effectiveness of each individual component.

However, a key limitation remains: the grouping mechanism is irreversible, meaning that once groups are formed, effective policies between groups cannot be shared. Future work will explore the use of graph neural networks [44] to establish an inter-group policy-sharing mechanism. Furthermore, we will investigate decoupled representation learning techniques to allow for the adoption of effective strategies from other groups while discarding suboptimal ones. The outcomes of this research highlight the significance of effective grouping, positioning our framework as a substantial contribution to advancing MARL.

Funding

The authors have nothing to report.

Conflicts of Interest

Xin Xu is a Deputy Editors-in-Chief for the journal, and was not involved in peer review process or the decision to publish this article. The authors declare that they have no conflict of interest.

Data Availability Statement

Given that this study focuses on the development and evaluation of a multi-agent reinforcement learning algorithm using simulated environments, no new datasets were generated or used. All experiments were conducted in publicly available simulated scenarios, including StarCraft II micromanagement tasks (SMAC) and Google Research Football (GRF) environments, which can be accessed via their official repositories.

References

1. Y. Ren, H. Zhang, L. Du, Z. Zhang, J. Zhang, and H. Li, "Stealthy Black-Box Attack With Dynamic Threshold Against MARL-Based Traffic Signal Control System," *IEEE Transactions on Industrial Informatics* 20, no. 10 (2024): 12021–12031, https://doi.org/10.1109/tii.2024.3413356.

2. T. Wu, P. Zhou, K. Liu, et al., "Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks," *IEEE Transactions on Vehicular Technology* 69, no. 8 (2020): 8243–8256, https://doi.org/10.1109/TVT.2020.2997896.

3. S. Morad, A. Shankar, J. Blumenkamp, and A. Prorok, "Language-Conditioned Offline RL for Multi-Robot Navigation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)* (2025), 14984–14991, https://doi.org/10.1109/ICRA55743.2025.11127288.

4. Y. Yang, Li Juntao, and P. Lingling, "Multi-Robot Path Planning Based on a Deep Reinforcement Learning DQN Algorithm," *CAAI Transactions on Intelligence Technology* 5, no. 3 (2020): 177–183, https://doi.org/10.1049/trit.2020.0024.

5. Y. Li, C. Yu, M. Shahidehpour, T. Yang, Z. Zeng, and T. Chai, "Deep Reinforcement Learning for Smart Grid Operations: Algorithms, Applications, and Prospects," *Proceedings of the IEEE* 111, no. 9 (2023): 1055–1096, https://doi.org/10.1109/JPROC.2023.3303358.

6. J. Wu and X. Xu, "Decentralised Grid Scheduling Approach Based on Multi-Agent Reinforcement Learning and Gossip Mechanism," *CAAI Transactions on Intelligence Technology* 3, no. 1 (2018): 8–17, https://doi.org/10.1049/trit.2018.0001.

7. J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to Communicate With Deep Multi-Agent Reinforcement Learning," in *Advances in Neural Information Processing Systems*, ed. D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Vol. 29 (Curran Associates, Inc, 2016), https://doi.org/10.48550/arXiv.1605.06676.

8. S. Xie, Y. Li, X. Wang, et al., "Hierarchical Relationship Modeling in Multi-Agent Reinforcement Learning for Mixed Cooperative–Competitive Environments," *Information Fusion* 108 (2024): 102318, https://doi.org/10.1016/j.inffus.2024.102318.

9. P. Sunehag, G. Lever, A. Gruslys, 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* (AAMAS, 2018), 2085–2087, https://doi.org/10.65109/jsrc7365.

10. T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning," in *Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research*, eds. J. Dy and A. Krause, Vol. 80 (PMLR, 2018), 4295–4304, https://doi.org/10.48550/arXiv.1803.11485.

11. 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 *Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research*, eds. K. Chaudhuri and R. Salakhutdinov, Vol. 97 (PMLR, 2019), 5887–5896, https://doi.org/10.48550/arXiv.1905.05408.

12. C. Li, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang, "Celebrating Diversity in Shared Multi-Agent Reinforcement Learning," in *Advances in Neural Information Processing Systems*, ed. M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, Vol. 34 (Curran Associates, Inc., 2021), 3991–4002, https://doi.org/10.48550/arXiv.2106.02195.

13. G. Papoudakis, F. Christianos, L. Schäfer, and S. V. Albrecht, "Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks," in *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)* (2021), https://doi.org/10.48550/arXiv.2006.07869.

14. Y. Yu, Q. Yin, J. Zhang, P. Xu, and K. Huang, "ADMN: Agent-Driven Modular Network for Dynamic Parameter Sharing in Cooperative Multi-Agent Reinforcement Learning," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24. International Joint Conferences on Artificial Intelligence Organization*, ed. K. L. Main Track (2024), 302–310, https://doi.org/10.24963/ijcai.2024/34.

15. J. Zhao, Y. Zhang, X. Hu, et al., "Revisiting Qmix: Discriminative Credit Assignment by Gradient Entropy Regularization," preprint, arXiv:2202.04427, February 09, 2022, https://doi.org/10.48550/arXiv.2202.04427.

16. Jiang, J. and Z. Lu, "The Emergence of Individuality," in *Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research. PMLR*, eds. M. Meila and T. Zhang. Vol. 139 (2021), 4992–5001, https://doi.org/10.48550/arXiv.2006.05842.

17. A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson, "MAVEN: Multi-Agent Variational Exploration," in *Advances in Neural Information Processing Systems*, ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Vol. 32 (Curran Associates, Inc, 2019), https://doi.org/10.48550/arXiv.1910.07483.

18. S. Liu, Y. Zhou, J. Song, et al., "Contrastive Identity-Aware Learning for Multi-Agent Value Decomposition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, no. 10 (2023), 11595–11603, https://doi.org/10.1609/aaai.v37i10.26370.

19. S. Li, J. K. Gupta, P. Morales, R. Allen, and M. J. Kochenderfer, "Deep Implicit Coordination Graphs for Multi-Agent Reinforcement Learning," in *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems. AAMAS '21. Virtual Event, United Kingdom* (International Foundation for Autonomous Agents and Multiagent Systems, 2021), 764–772: ISBN: 9781450383073, https://doi.org/10.65109/czoy2835.

20. W. Duan, J. Lu, Yu G. Wang, et al., "Layer-Diverse Negative Sampling for Graph Neural Networks," in *Transactions on Machine Learning Research* (2024): ISSN: 2835-8856, https://doi.org/10.48550/arXiv.2403.11408.

21. Y. Zang, J. He, K. Li, et al., "Automatic Grouping for Efficient Cooperative Multi-Agent Reinforcement Learning," *Advances in Neural Information Processing Systems* 36 (2023): 46105–46121, https://doi.org/10.52202/075280-1999.

22. H. Na, K. Lee, S. Lee, and I.-c. Moon, "Trajectory-Class-Aware Multi-Agent Reinforcement Learning," in *The Thirteenth International Conference on Learning Representations* (2025), https://doi.org/10.48550/arXiv.2503.01440.

23. T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson, and C. Zhang, "{RODE}: Learning Roles to Decompose Multi-Agent Tasks," in *International Conference on Learning Representations* (2021), https://doi.org/10.48550/arXiv.2010.01523.

24. Z. Hu, Z. Zhang, H. Li, C. Chen, H. Ding, and Z. Wang, "Attention-Guided Contrastive Role Representations for Multi-Agent Reinforcement Learning," in *The Twelfth International Conference on Learning Representations* (2024), https://doi.org/10.48550/arXiv.2312.04819.

25. T. Phan, F. Ritz, L. Belzner, P. Altmann, T. Gabor, and C. Linnhoff-Popien, "VAST: Value Function Factorization With Variable Agent Sub-Teams," in *Advances in Neural Information Processing Systems*, ed. A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (2021), https://dl.acm.org/doi/10.5555/3540261.3542100.

26. J. Shao, Z. Lou, H. Zhang, Y. Jiang, S. He, and X. Ji, "Self-Organized Group for Cooperative Multi-Agent Reinforcement Learning," in *Advances in Neural Information Processing Systems*, ed. S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Vol. 35 (Curran Associates, Inc., 2022), 5711–5723, https://doi.org/10.52202/068431-0413.

27. K. Kurach, A. Raichuk, P. Stańczyk, et al., "Google Research Football: A Novel Reinforcement Learning Environment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, no. 4 (2020), 4501–4510, https://doi.org/10.1609/aaai.v34i04.5878.

28. I. Mordatch and P. Abbeel, "Emergence of Grounded Compositional Language in Multi-Agent Populations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, no. 1 (2018), https://doi.org/10.1609/aaai.v32i1.11492.

29. B. Liao, C. Hua, Q. Xu, X. Cao, and S. Li, "Inter-Robot Management via Neighboring Robot Sensing and Measurement Using a Zeroing Neural Dynamics Approach," *Expert Systems With Applications* 244 (2024): 122938, https://doi.org/10.1016/j.eswa.2023.122938.

30. M. Liu, Y. Li, Y. Chen, Y. Qi, and L. Jin, "A Distributed Competitive and Collaborative Coordination for Multirobot Systems," *IEEE Transactions on Mobile Computing* 23, no. 12 (2024): 11436–11448, https://doi.org/10.1109/tmc.2024.3397242.

31. C. Hua, J. Xu, Z. Huang, B. Liao, and S. Li, "Optimization-Based Finite-Time Multi-Robot Formation: A Zeroing Neurodynamics

Method," *Tsinghua Science and Technology* 31, no. 1 (2026): 162–179, https://doi.org/10.26599/tst.2024.9010180.

32. T. Wang, C. Hua, Y. Wang, X. Cao, B. Liao, and S. Li, "Real-Time Formation Planning for Multirobot Cooperation: A Neural Informatics Perspective," *IEEE Transactions on Industrial Electronics* 72, no. 12 (2025): 13703–13715, https://doi.org/10.1109/TIE.2025.3579075.

33. T. Wang, H. Dong, V. Lesser, and C. Zhang, "ROMA: Multi-Agent Reinforcement Learning With Emergent Roles," in *Proceedings of the 37th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, Vol. 119 (PMLR, 2020), 9876–9886: Hal Daumé III and Aarti Singh, https://doi.org/10.48550/arXiv.2003.08039.

34. S. Fu, S. Zhao, T. Li, and Y. Yan, "QTypeMix: Enhancing Multi-Agent Cooperative Strategies Through Heterogeneous and Homogeneous Value Decomposition," *Neural Networks* 184 (2025): 107093, https://doi.org/10.1016/j.neunet.2024.107093.

35. Liu, C. and D. Li (2025), "HYGMA: Hypergraph Coordination Networks With Dynamic Grouping for Multi-Agent Reinforcement Learning," preprint, arXiv:2505.07207, May 12, 2025, https://doi.org/10.48550/arXiv.2505.07207.

36. F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems (Springer, 2016), https://doi.org/10.1007/978-3-319-28929-8.

37. H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning With Double Q-Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30, 2016), 1, https://doi.org/10.48550/arXiv.1509.06461.

38. M. Samvelyan, T. Rashid, C. S. de Witt, et al., "The StarCraft Multi-Agent Challenge," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (2019), https://doi.org/10.65109/lvzz5205.

39. J. Hu, S. Wang, S. Jiang, and M. Wang, "Rethinking the Implementation Tricks and Monotonicity Constraint in Cooperative Multi-agent Reinforcement Learning," in *The Second Blogpost Track at ICLR 2023* (2023), https://doi.org/10.48550/arXiv.2102.03479.

40. J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "{QPLEX}: Duplex Dueling Multi-Agent Q-Learning," in *International Conference on Learning Representations* (2021), https://doi.org/10.48550/arXiv.2008.01062.

41. B. Ellis, J. Cook, S. Moalla, et al., "SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning," in *Advances in Neural Information Processing Systems*, ed. A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Vol. 36 (Curran Associates, Inc., 2023), 37567–37593, https://doi.org/10.52202/075280-1634.

42. H. Zhang, H. Yu, X. Wang, et al., "Knowledge-Guided Communication Preference Learning Model for multi-agent Cooperation," *Information Sciences* 667 (2024): 120395, https://doi.org/10.1016/j.ins.2024.120395.

43. Z. Xie, C. Ji, C. Qiao, et al., "Mutual Information Oriented Deep Skill Chaining for multi-agent Reinforcement Learning," *CAAI Transactions on Intelligence Technology* 9, no. 4 (2024): 1014–1030, https://doi.org/10.1049/cit2.12322.

44. W. Duan, J. Lu, and J. Xuan, "Group-Aware Coordination Graph for Multi-Agent Reinforcement Learning," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, ed. K. L. Main Track (International Joint Conferences on Artificial Intelligence Organization, 2024), 3926–3934, https://doi.org/10.24963/ijcai.2024/434.