

Learning to Collaborate by Grouping: A Consensus-Oriented Strategy for Multi-Agent Reinforcement Learning

Jingqing Ruan^a, Xiaotian Hao^b, Dong Li^{c,*} and Hangyu Mao^d

^aInstitute of Automation, Chinese Academy of Sciences

^bTianjin University

^cNoah's Ark Lab, Huawei

^dSenseTime Research

Abstract. Multi-agent systems require effective coordination between groups and individuals to achieve common goals. However, current multi-agent reinforcement learning (MARL) methods primarily focus on improving individual policies and do not adequately address group-level policies, which leads to weak cooperation. To address this issue, we propose a novel *Consensus-oriented Strategy* (CoS) that emphasizes group and individual policies simultaneously. Specifically, CoS comprises two main components: (a) the vector quantized group consensus module, which extracts discrete latent embeddings that represent the stable and discriminative group consensus, and (b) the group consensus-oriented strategy, which integrates the group policy using a hypernet and the individual policies using the group consensus, thereby promoting coordination at both the group and individual levels. Through empirical experiments on cooperative navigation tasks with both discrete and continuous spaces, as well as google research football, we demonstrate that CoS outperforms state-of-the-art MARL algorithms and achieves better collaboration, thus providing a promising solution for achieving effective coordination in multi-agent systems.

1 Introduction

Many applications, such as multi-player games [17], and sensor networks [40], can be modeled as cooperative multi-agent systems (MASs), where a team of agents performs a shared task to reach a common goal. A promising solution is cooperative multi-agent reinforcement learning (MARL) which has shown exceptional results, of which the popular approach for multi-agent cooperation is communication-based MARL. One line of research lies on fine-grained communication channels, such as graph neural networks [14, 25], attention mechanisms [20, 15], etc. Another line [4, 21] yields agents that learn communication protocols to determine which messages to transmit, and who to communicate with to assist decision-making. However, a common issue is a lack of reasonable labor division for multiple agents.

In order to achieve better collaboration, many realistic multi-agent problems require a reasonable division of labor to enhance team-level cooperation [6, 2]. For example, in search and rescue missions, multiple agents (e.g. drones, robots, doctors) need to coordinate their

efforts to locate and rescue survivors in a disaster-stricken area. Multiple drones should be deployed to search the area to ensure complete coverage of the search area and avoid collisions, while the doctors and robots should be assigned dynamically to different areas to ensure efficient rescuing. Thus, a well-designed cooperative multi-agent system can dynamically group agents with similar abilities based on the specific situation, while also allowing individual agents to make rational decisions based on their specific observations.

Some methods have been proposed from the perspective of roles or groups. ROMA [36] and RODE [35] focus on task decomposition and specialize the agent associated with a role to resolve a certain sub-task. LDSA [38] learns dynamic subtask assignments, which can dynamically group agents with similar abilities into the same sub-task. However, on the one hand, role (group) representations are not well constrained and vulnerable to dynamic changes from the environment or policy training. On the other hand, the above methods rarely consider extracting higher-level policy guidance from the natural properties of groups (sub-tasks). In summary, good teamwork often requires a good division of labor, forward-looking guidance to specialize in a certain subtask, and rational individual decision-making. This poses a challenge for MARL for providing stable and powerful group representations, higher guidance at the team level, and better decisions at the individual level.

Therefore, we propose CoS, a consensus-oriented strategy in MARL. First, we use the vector quantized variational autoencoder (VQ-VAE) [34] to extract the group consensus embedding, which captures the shared objective that agents in the same group should pursue and is essential for promoting effective teamwork. Then, we perform the policy learning from two levels. At the higher level, we propose using a hyper-network architecture [11] to transform group consensus embedding into group-level decisions from global and long-term perspectives. At the lower level, we utilize the group consensus embedding as the context prompt to augment the observation to make the individual and refined policy. The combination of these two policies guarantees the foresight and precision of the decision-making process. Moreover, CoS is a pluggable module and is suitable for both discrete and continuous action spaces. We evaluate our method in three challenging MARL environments including discrete cooperative navigation, continuous cooperative navigation [23], and google research football [17]. The results show that our CoS sig-

* Corresponding Author. Email: lidong106@huawei.com

nificantly improves the learning performance on these benchmarks compared to some competitive baselines.

2 Related Work

Multi-agent Grouping. We classify existing algorithms into three categories: (i) prior knowledge-based, (ii) role-based, and (iii) subtask-based methods. Some early methods [26, 18, 7, 27] utilize the domain knowledge to group the agents, and are limited by human labor. To solve the issue, one line of work groups multiple agents by generating diverse roles responsible for different parts. ROMA [36] learns a role-specific policy where the roles are captured from agents' local observations. ROGC [22] introduces the graph convolutional network for classifying agents into different roles. RODE [35] decomposes action spaces with the learned roles. [12] define the role and role diversity to measure a cooperative MARL task and help diagnose the current policy. LILAC [9] learns a leader to assign roles. Another line of work such as [38, 39, 28, 13], divides the agents into some groups that carry out similar sub-tasks with a specific policy or value function. In our work, we learn more stable and distinguishable group embeddings and further consider the integration of team-level strategy and individual-level decision.

Representation Learning in MARL. The representations for observations, actions, or underlying messages are widely studied in MARL. Some works [3, 8] use bisimulation metrics to extract the latent embeddings from observations. [19, 1, 31, 30] attempt to learn action representations to assist multi-agent policy learning. [37, 16] propose represent underlying messages to conduct effective communication in MAS. There are also some works like [32, 41] that use VAE to encode the trajectory message to make representation more knowledgeable. Unlike the above methods, we introduce VQ-VAE to maintain a quantized hidden space to extract stable, and distinguishable group consensus embeddings to associate a more powerful strategy.

3 Preliminary

Vector Quantised-Variational AutoEncoder. The VQ-VAE is a type of variational autoencoder that uses vector quantization to obtain a discrete latent representation. It includes an encoder z_e , a decoder z_d , and a codebook $e \in \mathbb{R}^{K \times D}$, where K is the number of embeddings in the codebook and D is the dimension of the embeddings. The encoder maps the input data x to a sequence z of discrete codes from a codebook and the decoder takes the responsibility of reconstruction. In our paper, we use the encodings in the codebook as stable and distinguishable representations.

Formally, $z_e(x)$ is mapped via nearest-neighbor into the quantized codebook, denoted as:

$$z_q(x) = e^j \text{ where } j = \operatorname{argmin}_k \|z_e(x) - e^k\|_2^2. \quad (1)$$

This discretized process is not differentiable, thus copying the gradient of $z_q(x)$ to $z_e(x)$ is a suitable approximation similar to the straight-through estimator. The loss can be written as follows.

$$\mathcal{L} = \|z_d(z_q(x)) - x\|_2^2 + \beta \|sg[z_e(x)] - e^j\|_2^2 + \|z_e(x), sg[e^j]\|_2^2, \quad (2)$$

where sg is a stop gradient operator, and β is a parameter that regulates the rate of code change.

4 Methodology

4.1 Problem Setup

Group Consensus-guided Dec-POMDP. We formalize our problem with multiple agents as a group consensus-guided decentralized partially observable Markov decision process (GC-Dec-POMDP). A GC-Dec-POMDP is given by a tuple $(\mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{G}, \mathcal{A}, P, R, \gamma)$. \mathcal{I} denotes the set of agents indexed by $i \in 1, \dots, N$. Let \mathcal{S} represent the global state space and \mathcal{A} be the action space of the agent. $\mathcal{G} = \{e_1, e_2, \dots, e_K\}$ denotes the group codebook to generate the group consensus embedding e^j belonging to the group j for each agent i in every time step, which divides all the agents into various parts to realize the corresponding density of coordination. At each time step t , each agent i chooses its action $a_t^i \in \mathcal{A}$ based on the group consensus policy π^{gc^j} conditioned only on the group consensus embedding e^j and the group guided policy π^{gg^j} conditioned on e^j and its observation drawn from the observation function $O(s_t, i)$ where $s_t \in \mathcal{S}$, and receive the environmental reward r_t given by its reward function $R^i : \mathcal{O} \times \mathcal{A} \rightarrow \mathbb{R}$. $P(s_{t+1}|s_t, \mathbf{a}_t)$ is the dynamics function that gives the distribution of the next state s_{t+1} at the current state s_t to execute the joint action $\mathbf{a}_t = \{a_t^i\}_{i=1}^N$. In summary, each agent learns the group consensus policy π^{gc^i} and the group guided policy π^{gg^i} simultaneously, denoted as $\pi^i = (\pi^{gc^i}, \pi^{gg^i})$, and the overall objective is to find the optimal joint policy $\pi = (\pi_1, \dots, \pi_N)$ such that the discounted returns of each agent $G_i = \sum_{t=1}^{\infty} \gamma^{t-1} r_t^i$ are maximized, where $\gamma \in [0, 1)$ is a discounted factor.

Optimization Target. The overall goal is to maximize the cumulative return, denoted as:

$$\eta = \mathbb{E}_{\mathbf{a} \sim \pi} \left[\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, \mathbf{a}_{t+k}) \right], \quad (3)$$

where π is the joint policy for all the agents, formulated as:

$$\pi(\mathbf{a}|\{o^i\}_{i=1}^N) \triangleq \{\pi^{gc^i}(a^{gc^i}|e^j) + \pi^{gg^i}(a^{gg^i}|o^i, e^j)\}_{i=1}^N \quad (4)$$

where e^j is the group consensus embedding. Here, we drop the time step t for simplification.

4.2 The Motivation and Framework

Effective collaboration is a crucial aspect of multi-agent systems and relies heavily on well-coordinated teamwork. One approach to achieving effective collaboration is through the use of consensual groups, which consist of one or more agents guided by a shared pattern to make higher-level decisions for completing a particular sub-task. To ensure the success of the consensual groups, we consider three fundamental principles:

a) **Group Consensus:** Agents within a group should strive to reach a consensus to accomplish a specific sub-task. The group consensus should be well-defined and distinguishable, with sufficient knowledge.

b) **Behavioral Diversity:** Both inter- and intra-group agents should exhibit diverse and varied behavioral patterns to promote exploration and better solutions.

c) **Dynamic Integration:** The multi-agent decision-making process should involve a dynamic combination of a group consensus policy and an individual policy guided by group knowledge. This

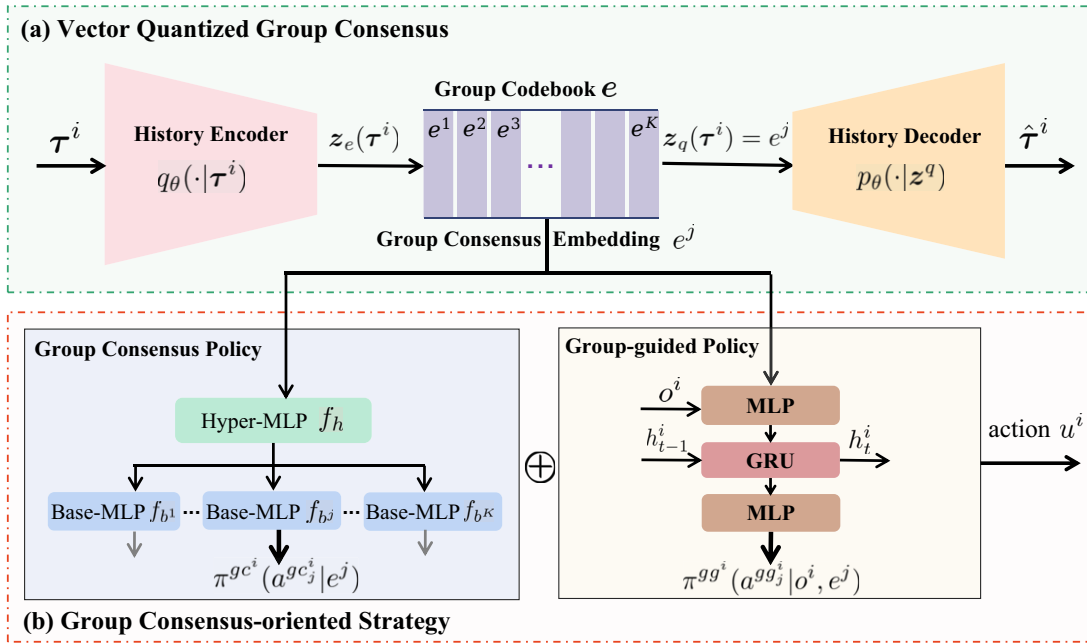


Figure 1: The schematics of our framework CoS. CoS includes two parts: (a) The extraction of group consensus embeddings is finished by the module of vector quantized group consensus. (b) The group consensus-oriented strategy is utilized to generate the overall decisions.

approach will allow for flexibility in adapting to changing circumstances and enable the agents to make informed decisions in a collaborative manner.

Taking the above factors into consideration, we present the Consensus-oriented Strategy (CoS) learning framework, shown in Figure 1. The upper part is designed for extracting knowledgeable and distinguishable group consensus embedding. The lower part illustrates the group consensus-oriented strategy. Compare to common MARL methods, we innovatively introduce the VQ-VAE for learning group consensus embedding and propose an additional group consensus policy. This embedding module can be considered as observation augmentation, and the group consensus policy can be incorporated into most MARL algorithms to complement their policy. Moreover, the network architecture and the detailed parameter settings can be found in Appendix A.1 and A.2, respectively. The pseudo-code for CoS is provided in A.3¹.

4.3 Vector Quantized Group Consensus

Drawing inspiration from the vector-quantized variational autoencoder (VQ-VAE) proposed by [34], we introduce a novel module called Vector Quantized Group Consensus (VQGC) to achieve identifiable and rich group consensus. The motivations behind this include: a) The discrete codebook in VQ-VAE can effectively group features with similar semantics into tighter clusters with better separation than traditional continuous representation methods. This enables more robust and efficient feature extraction in multi-agent settings. b) The range of VQ encodings is highly controllable, and training the VQ encodings as group embeddings of agents leads to reduced variance, thus enhancing the stability of multi-agent collaboration. c) The use of a discrete feature representation can mitigate the effects of semantic noise in stochastic environments by limiting the number of possible bias vectors. This, in turn, ensures the extraction of richer knowledge from the features.

Thus, building on the powerful properties of VQ-VAE, we design the VQGC to extract the group consensus embeddings, which enables neural models to learn similarities and differences between states better and generate a higher-level consensual representation to guide policy learning. Next, the design details are elaborated as follows.

History Encoder. At time step t , given the history transitions τ^i including observations o^i , action a^i , and reward r^i , we first aim to extract the feature by a history encoder parameterized by θ , denoted as:

$$z_e(\tau^i) = q_\theta(\cdot|\tau^i), \quad (5)$$

where $\tau^i := \{o_t^i, a_{t-1}^i, r_{t-1}^i\}_{i=t-c}^t$ and c is the window size of the history chunk.

Group Codebook. The group codebook refers to a set of vectors used in the VQGC module. Specifically, it is defined as $\mathbf{e} = \{e^1, e^2, \dots, e^K\}, \forall e^j \in \mathbb{R}^{1 \times D}$, where j is the j^{th} entry and D is the dimensionality of each entry. To obtain a powerful and robust group codebook, we introduce a regularizer inspired by [5] into the training process. Euclidean space has limited representation capacity in a fixed dimension, and increasing the dimensionality will bring a large computation budget and training overfitting issues. Thus, we consider a hyperbolic space constraint to generate more powerful and knowledgeable embedding representations. Thus, we give a definition of the Poincaré ball model [29].

Definition 1 *The Poincaré ball model is a model of n -dimensional ($n \geq 3$) hyperbolic geometry in which the points of the geometry are in the n -dimensional unit ball.*

Let a Poincaré ball with dimension d and radius 1 be $\mathcal{P}^{d,1} := \{e \in \mathbb{R}^d, \|e\| < 1\}$, and the operator $\|\cdot\|$ denotes the Euclidean L^2 norm. This corresponds to the Riemannian manifold $(\mathcal{P}^{d,1}, g_e)$, where $g_e = (2/(1-\|e\|^2))^2 g^E$ is the Riemannian metric tensor and

¹Appendix is at: <http://arxiv.org/abs/2307.15530>

g^E denotes the Euclidean metric tensor. Then the distance between two vector $\mathbf{x}, \mathbf{y} \in \mathcal{P}^{d,1}$ can be computed as:

$$d(\mathbf{x}, \mathbf{y}) = \operatorname{arcosh} \left(1 + 2 \frac{\|\mathbf{x} - \mathbf{y}\|^2}{(1 - \|\mathbf{x}\|^2)(1 - \|\mathbf{y}\|^2)} \right). \quad (6)$$

For inducing an appropriate structural bias on the group embedding space, we introduce a novel metric $\mathcal{L}_{\mathcal{P}}$ to constrain the embeddings into hyperbolic space with the Poincaré ball model, which is well-suited for the gradient-based optimization [24], formulated as:

$$\mathcal{L}_{\mathcal{P}} = \sum_{(e^I, e^j) \in \mathcal{B}_e} \log \sigma(\operatorname{Sgn}((e^I, e^j)) \cdot D(e^I, e^j)), \quad (7)$$

where $D(e^I, e^j) = \min_{i \in I} d(e^i, e^j)$ is the shortest Poincaré distance for the anchor e^j and other entries e^I sampled from the buffer \mathcal{B}_e that is used to save the latest L group codebooks. This buffer at time step t is denoted as $\mathcal{B}_t^e = \{e_t^1, e_t^2, \dots, e_t^K\}_{t=t-L}^t$. σ is a logistic activation function and Sgn is a symbolic function defined as follows.

$$\operatorname{Sgn}((e^I, e^j)) = \begin{cases} 1, & \text{if } e^I \in \operatorname{Pos} \\ -1, & \text{if } e^I \in \operatorname{Neg} \end{cases} \quad (8)$$

where the set Pos denotes the identical group embedding e_i^i with different time steps. The set Neg represents the different group embeddings e_i^j , where $j \neq i, \forall i \in (t-L, t]$.

By minimizing $\mathcal{L}_{\mathcal{P}}$, we can pull together the group embeddings with the same identifier to relieve the non-stationarity of drastic changes and push apart different group embeddings to obtain distinguishable representations.

Given the current group codebook e and the encoded feature $z_e(\tau^i)$, the nearest matching mechanism can be denoted as *Quantize*, formulated as follows.

$$\operatorname{Quantize}(z_e(\tau^i)) = e^j, \quad j = \operatorname{argmin}_k \|z_e(\tau^i) - e^k\|, \quad (9)$$

where k is the index of the length of the group codebook. In the following, we denote Equation (9) as e^j to represent the nearest quantized vector for brevity.

History Decoder. The history decoder takes the $z_q(\tau^i)$ as the input to reconstruct the input τ^i that is to maximize the log-likelihood:

$$\mathcal{L}_{\operatorname{recon}} = \mathbb{E}_{z_q} \left[\log p(\hat{\tau}^i | z_q(\tau^i)) \right]. \quad (10)$$

Thus, the decoding of the history transitions can capture the dynamics of the environment to a large extent.

Training Objective. Based on VQVAE, the training of VQGC is equipped with the stop gradient technique, summarized as follows.

$$\mathcal{L}_{VQGC} = \underbrace{\log p_{\theta}(\hat{\tau}^i | z_q(\tau^i))}_{\text{history decoder}} + \beta \underbrace{\|sg[z_e(\tau^i)] - e^j\|_2^2}_{\text{history encoder}} + \underbrace{\|z_e(\tau^i), sg[e^j]\|_2^2 + \mathcal{L}_{\mathcal{P}}}_{\text{group codebook}}, \quad (11)$$

where the operator sg indicates the stop of gradient backpropagation. β is the hyper-parameter preventing the encoder outputs from fluctuating between different code vectors.

So far, the group consensus embedding is outputted by our VQGC. Through training, the group consensus embedding fully incorporates rich dynamical knowledge, as well as robust and distinguishable representations. Next, we will introduce the group consensus-oriented strategy to use this extracted knowledge to provide global intention and individual guidance for the multi-agent decision-making process.

4.4 Group Consensus-oriented Strategy

Our group consensus-oriented strategy includes two parts: group consensus policy aims to generate the higher-level decision from the global and long-term perspective and group-guided policy is used to make individual decisions condition on the group consensus, elaborated as follows.

Group Consensus Policy (GCP). Here, we propose using a hyper-network architecture where a primary network f_h generates the weights to parameterize all the base layer f_b^i , where $i \in \{1, 2, \dots, K\}$. A hyper-network is a network that generates the weights for another network. Specifically, VQGC generates the group consensus embedding e^j for each agent i . Then the hyper-MLP takes the group consensus embeddings e^j as input and produces the weights for each Base-MLP network which performs fine-grained control denoted as $f_{b^j} = \pi^{gc^i}(a^{gc^i} | e^j)$. The hyper-MLP can be formulated as $f_h : \mathbb{R}^d \mapsto \mathbb{R}^{\dim(b^j)}$ which is parameterized by h, d and $\dim(b^j)$ are the dimensions of group embedding and base network, respectively. The formulation of a Base-MLP f_{b^j} is given by $f_{b^j}(e^j) = \mathbf{W}_1(\sigma(\mathbf{W}_2^T e_j^T))$, where the weight matrix \mathbf{W}_1 and \mathbf{W}_2 are produced by Hyper-MLP f_h .

The motivation for such a design centrally involved two aspects. On the one hand, agents from different groups are expected to solve different sub-tasks on different parameter spaces, which escalates the difficulties of policy learning. The hyper-network fed with different group consensus embeddings can well coordinate the potentially conflicting parameters in a unified space, which reduces the training complexity to some extent. On the other hand, the hyper-network captures higher-order interaction among group consensus embeddings, which is conducive to making global decisions.

Group-Guided Policy (GGP). Besides the higher-level decision, the individual policy guided by the group that emphasizes the underlying dynamic of the environment is also essential. Here, we utilize the group consensus embedding e^j as the context prompt to augment the observation o^i for agent i belonging to the group j , denoted as $\pi^{gg^i}(a^{gg^i} | o^i, e^j)$. The group-guided policy is used to make individual decisions conditioned on the group consensus.

Training Objective. Borrow the policy gradient for perturbation network [10], the final executed action of the agent i can be denoted as follows.

$$u^i = \pi^{gc^i}(e^j) + \pi^{gg^i}(o^i, e^j) \quad (12)$$

With Equation (3) and (12), the overall optimization objective for the expected cumulative return can be written as:

$$\mathcal{J} = \mathbb{E}_{s, e^j, \mathbf{u}} \left[\prod_{i=1}^N \left(\pi^{gc^i}(e^j) + \pi^{gg^i}(o^i, e^j) \right) \cdot Q_{\pi}(s, \mathbf{u}) \right], \quad (13)$$

where $s \sim p^{\pi}$ is the state sampled from the stationary distribution p^{π} , $e^j \sim z_q$ is the generated group embedding, and $\mathbf{u} \sim \{\pi^{gc^i} + \pi^{gg^i}\}_{i=1}^N$ denotes the joint action.

Thus, the gradient for the group consensus policy of agent i parameterized by ϕ can be derived by applying the mini-batch technique to the off-policy training:

$$\nabla_{\phi}^i \mathcal{J} = \mathbb{E} \left[\nabla_{\mathbf{u}'} Q_{\pi}(s, \mathbf{u}') \Big|_{\mathbf{u}' = \{\pi_{\phi}^{gc^i} + \pi_{\phi}^{gg^i}\}} \nabla_{\phi} \log \pi_{\phi}^{gc^i}(e^j) \right]. \quad (14)$$

Table 1: End steps of various methods on d-CN.

#agents	CoS	MAPPO	VDN	QMIX	RODE	ROMA
4	19.8 \pm 1.1	21.2 \pm 0.7	24.6 \pm 0.4	24.3 \pm 0.3	23.6 \pm 1.2	24.8 \pm 0.9
6	23.1 \pm 0.9	23.2 \pm 0.8	24.9 \pm 0.6	24.7 \pm 0.5	24.9 \pm 0.8	24.9 \pm 0.6
10	25 \pm 0	25 \pm 0	25 \pm 0	25 \pm 0	25 \pm 0	25 \pm 0

Also, the gradient for the group-guided policy is shown as:

$$\nabla_{\phi}^i \mathcal{J} = \mathbb{E} \left[\nabla_{\mathbf{u}'} Q \pi(s, \mathbf{u}') \Big|_{\mathbf{u}' = \{\pi_{\phi}^{gc^i} + \pi_{\phi}^{gg^i}\}} \nabla_{\phi} \log \pi_{\phi}^{gg^i}(o^i, e^j) \right]. \quad (15)$$

Jump-start Dynamic Integration. However, evidence from supervised learning suggests hypernetwork performance is highly sensitive to input and initialization. Thus, we perform a trick called Jump-start Dynamic Integration (JDI) to alleviate the issue inspired by [33]. Here, we define a hyper-parameter $jump \in [0, 1]$ to represent a proportion factor of the full training horizon T , which indicates it should be changed while the current time step t satisfies $t \geq jump * T$. The change can be formulated as:

$$\pi^{final} = \begin{cases} \pi^{gg^i}(o^i, e^j), & \text{if } t < jump * T \\ \pi^{gc^i}(e^j) + \pi^{gg^i}(o^i, e^j), & \text{else} \end{cases}. \quad (16)$$

In our case, we adopt this trick to stabilize the training process of our group consensus policy by warming up the training of the group codebook.

5 Experimental Evaluations

We evaluate the effectiveness of our algorithm in three environments including both discrete and continuous action space: discrete Cooperative Navigation (d-CN), continuous Cooperative Navigation (c-CN), and Google Research Football (GRF), illustrated as Figure 2.

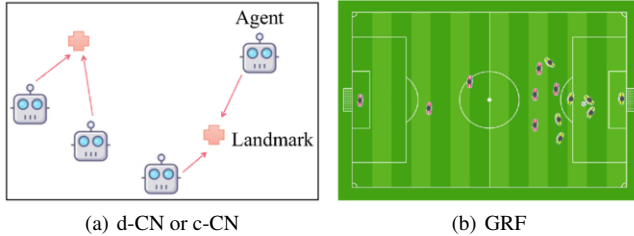


Figure 2: The schematics of our experimental environments.

5.1 Experimental Settings

d-CN and c-CN. We modify the classic Cooperative Navigation (CN) implemented in the multi-agent particle world [23] to a more challenging environment, which requires more collaboration among agents. We initialize the CN world with n landmarks and $2 * n$ agents with random locations at the beginning of each episode. Each agent can only observe its velocity, position, and displacement from other agents and landmarks. The final objective is to occupy all the landmarks and each landmark contains two agents. The reward function can be formulated as $r_t = -0.1 + 3 * single + 10 * double$. The variables *single* and *double* denote the number of landmarks that are occupied by only one and two agents, which corresponds to the reward 3 and 10, respectively. -0.1 is the step punishment. Obviously, the game can reach the maximum reward while each landmark contains two agents, and can be over. d-CN denotes that the world has a discrete action space \mathcal{A}_d including five actions [*up*, *down*, *left*, *right*, *stop*]. c-CN represents that the world has a continuous action space \mathcal{A}_c . We set the length of each episode as 25 time steps.

GRF. GRF [17] is a realistically complicated and dynamic multi-agent testbed. Agents should have a division of labor and plan to coordinate the time and location to complete the scoring. In the experiments, we control left-side players except for the goalkeeper while the right-side players are built-in bots controlled by the game engine. Here, each player has 19 actions to control, including the standard move actions and different ball-kicking techniques. The observation contains the positions and moving directions of the ego-agent, other agents, and the ball. We use the Floats wrappers to represent the state that contains a 115-dimensional vector. The rewards include the SCORING reward $\{-1, +1\}$, and the CHECKPOINT reward, which is the shaped reward that specifically addresses the sparsity of SCORING. The detailed descriptions of GRF can be found in Appendix A.4.

Baselines. We compared our results with several baselines as follows. VDN and QMIX are state-of-the-art (SOTA) value factorization approaches, with which it is difficult to obtain coordinated behaviors. MAPPO is the multi-agent competitive SOTA algorithm extended from PPO by setting the sharing actor and a centralized critic. MADDPG and MATD3 are classic continuous control algorithms. ROMA and RODE are role-based grouping algorithms. Note that VDN, QMIX, ROMA, and RODE are designed for the discrete action space, while MADDPG and MATD3 are designed for the continuous action space.

5.2 Does CoS Perform Better?

For validating our CoS, we conduct empirical experiments on d-CN, c-CN, and GRF. The benchmarks we choose include discrete and continuous action spaces, realistically complicated and stochastic worlds. All experiments are repeated for 10 runs with different random seeds.

5.2.1 Performance on d-CN.

We conduct the experiments across 4, 6 and 10 agents in d-CN with discrete action space as shown in Figure 3(a-c). CoS substantially gets a better average reward than all the baselines, indicating that the group consensus strategy increasingly enhances the superiority of our method.

Moreover, as shown in Table 1, we report the mean end steps of these methods after testing 100 episodes, and the episode limit is set to 25. CoS basically completes the task faster with the least number of steps. These results show that the group consensus strategy of CoS can group agents with different intentions toward various landmarks, which boosts the training performance.

5.2.2 Performance on c-CN.

As shown in Figure 3(d-f), the results in continuous space also exhibit better performance. Obviously, our method has a quicker convergence speed and a smaller variance than others, which indicates the decision of CoS to consider both global and individual guidance

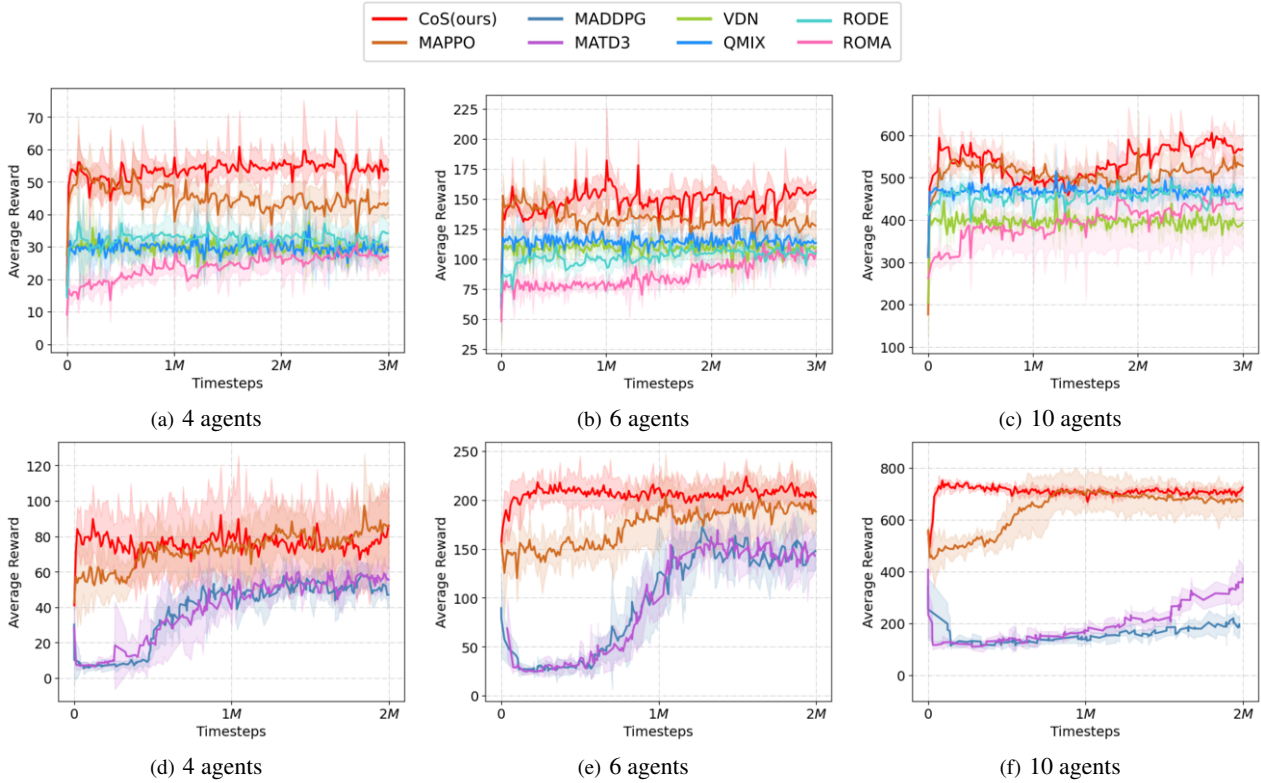


Figure 3: Average episodic rewards and the confidence level for 4, 6, and 10 agents on CN. (a-c) The results on d-CN. (d-f) The results on c-CN.

can further exploit the underlying properties and facilitate cooperative behaviors among agents.

In Table 2, our method still get the least number of steps. Unfortunately, all the methods have failed with 10 agents because of the stochasticity and difficulty of the environment. It will be an interesting direction to study how to obtain the optimal solution in such complicated scenarios in the fastest time steps.

Table 2: End steps of various methods on c-CN.

#agents	CoS	MAPPO	MADDPG	MATD3
4	19.6 ± 2.1	20.3 ± 2.3	22.3 ± 1.2	22.1 ± 1.6
6	22.1 ± 1.4	23.8 ± 1.5	24.3 ± 1.3	24.2 ± 1.3
10	25 ± 0	25 ± 0	25 ± 0	25 ± 0

5.2.3 Performance on GRF.

Further, we conduct experiments in google research football, a more dynamic and complicated benchmark to validate the effectiveness of our proposed CoS, shown in Figure 4. Specifically, we choose three popular scenarios, including *academy_3_vs_1_with_keeper* (3vs1), *academy_counterattack_easy* (c_easy), and *academy_counterattack_hard* (c_hard).

In Figure 4(a-c), we observed that CoS consistently obtains higher performance than all the baselines in different scenarios of GRF, indicating that our method is robust and effective in complex and dynamic environments. Moreover, this performance improvement in GRF demonstrates that our approach excels at handling stochasticity, as the dynamic grouping strategy can extract more underlying details and make informed high-level decisions. Additionally, we report the average end steps of CoS in the three scenarios, shown in Figure 4(d). We test the trained model for 100 episodes and count the average end steps of every algorithm. Specifically, we tested the trained model for

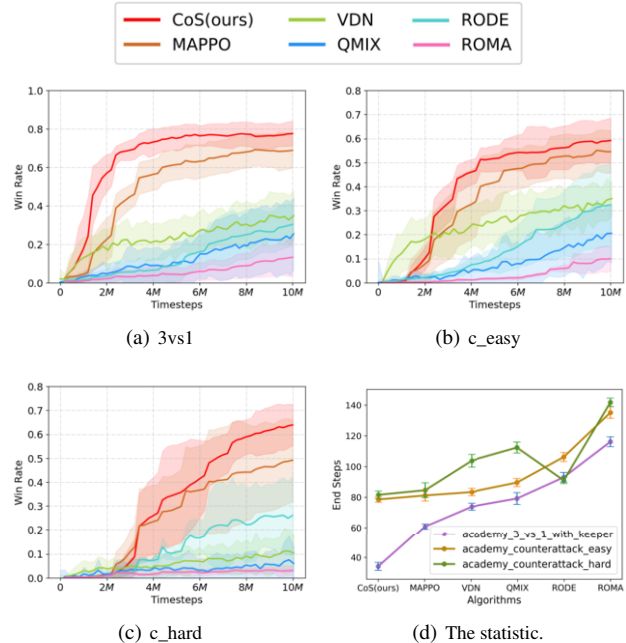
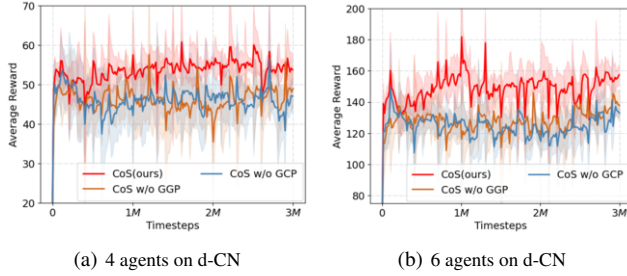


Figure 4: The performance on three scenarios of GRF.

100 episodes and counted the average end steps of each algorithm. Notably, CoS completed the football game in the fewest steps compared to all the baselines, further validating the superiority of our proposed algorithm.



(a) 4 agents on d-CN (b) 6 agents on d-CN

Figure 5: The ablation study on d-CN of 4, 6 agents.

5.3 Does the components of CoS Really Work?

To evaluate the effectiveness of the components in CoS, we conduct the ablation study with the following configurations.

- *CoS(ours)*: The proposed framework.
- *CoS w/o GGP*: Remove the group-guided policy.
- *CoS w/o GCP*: Remove the group consensus policy.

As shown in Figure 5, the ablation study of 4 and 6 agents on d-CN shows the components in the CoS really work. The performance degradations of *CoS w/o GGP* and *CoS w/o GCP* show that the combination of these two modules is necessary, mainly due to two aspects. (1) GCP utilizes the extracted group consensus embeddings to make high-level decisions, which is conducive to long-term utility. (2) GGP perceives the individual observation from the true world, which can exploit underlying dynamics to make accurate decisions. Therefore, the fine-grained combination of these two modules can achieve forward-looking guidance to specialize in a certain subtask and rational individual decision-making.

5.4 How Well Does the Group Consensus Embedding Perform?

To investigate the discriminative power of the CoS model’s group consensus embeddings, we visualize the embeddings of several scenarios on two tasks collected in the later stages of the training process. The goal is to show whether CoS can distinguish different groups and achieve separate consensus among every group, aiding in the collaboration of intra-groups in the decision-making process.

Due to the assumption that K agents can group into K groups maximally, the number of group clusters is the same as the number of agents in each scenario. As shown in Figure 6, the 3D visualizations demonstrate that CoS can consistently produce distinct group embeddings in all the scenarios. We showcase the visualizations obtained by applying the T-SNE technique to the group consensus embeddings saved from the last 5000 training steps. Each cluster in the plots represents a specific group. Each point in the cluster corresponds to a vector in the group codebook. The distinguishable border highlights the ability of CoS to learn consistent group consensus representations.

Overall, these results demonstrate that our proposed method can facilitate intra-group collaboration and decision-making by producing effective group consensus embeddings that can discriminate between different groups in the decision-making process.

5.5 Has CoS Learned to Group?

To assess CoS’s ability to group and its impact on the collaborative behavior of agents, we conducted some visualizations and analyses. First, at time step $t = 20$ during one testing episode, we visualize

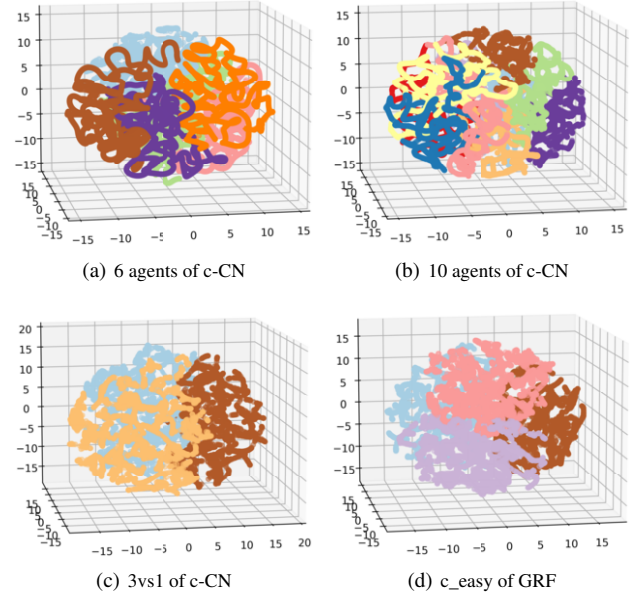


Figure 6: Visualization of the group consensus embeddings on c-CN and GRF. We sample 5000 points for each group.

the grouping effect, as shown in Figure 7(a), which illustrates that agents grouped by the same number in yellow exhibit similar behavior, indicating that CoS has learned to divide the agents into groups and promote collaboration.

To further examine the effect of grouping on performance, we evaluate the average rewards of 100 episodes at different checkpoints with varying numbers of groups, as shown in Figure 7(b). Our results show that in the scenario with 10 agents, dividing them into 5 groups achieves the highest utility, consistent with the environmental setting. This finding highlights the importance of an effective grouping mechanism and demonstrates the capability of CoS to optimize the group sizes based on the task requirements.

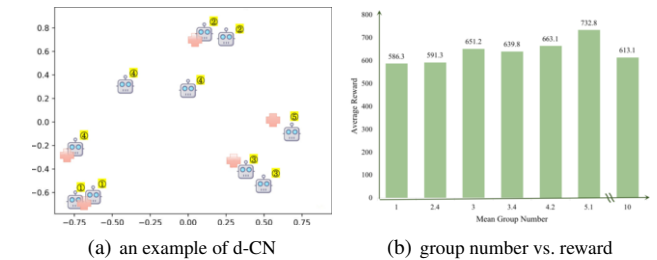


Figure 7: Visualization of Grouping Effects.

6 Conclusions

In this paper, we propose CoS, a novel consensus-oriented strategy to promote multi-agent collaboration. First, CoS leverages the vector quantized variational autoencoder to extract the distinguishable and stable group consensus embeddings. Furthermore, the embeddings are used to assist global and individual decisions through the proposed group consensus policy and group-guided policy, respectively. Our empirical results on three benchmarks show that CoS significantly outperforms existing state-of-the-art methods. Additionally, the ablation study demonstrates the necessity of combining the two policies, and the visualizations further validate the effectiveness of CoS in promoting collaboration among agents. In the future, it is an interesting direction to investigate universal group embeddings across different tasks.

References

- [1] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare, 'Contrastive behavioral similarity embeddings for generalization in reinforcement learning', *arXiv preprint arXiv:2101.05265*, (2021).
- [2] Md Golam Rabiul Alam, Yan Kyaw Tun, and Choong Seon Hong, 'Multi-agent and reinforcement learning based code offloading in mobile fog', in *2016 International Conference on Information Networking*, pp. 285–290, (2016).
- [3] Bogdan Aman and Gabriel Ciobanu, 'Knowledge dynamics and behavioural equivalences in multi-agent systems', *Mathematics*, **9**(22), 2869, (2021).
- [4] Yali Du, Bo Liu, Vincent Moens, Ziqi Liu, Zhicheng Ren, Jun Wang, Xu Chen, and Haifeng Zhang, 'Learning correlated communication topology in multi-agent reinforcement learning', *International Conference on Autonomous Agents and Multi-Agent Systems*, (2021).
- [5] Yue Fan and Xiuli Ma, 'Multi-vector embedding on networks with taxonomies', (2021).
- [6] Jacques Ferber and Olivier Gutknecht, 'A meta-model for the analysis and design of organizations in multi-agent systems', in *Proceedings international conference on multi agent systems (Cat. No. 98EX160)*, pp. 128–135, (1998).
- [7] Jacques Ferber, Olivier Gutknecht, and Fabien Michel, 'From agents to organizations: an organizational view of multi-agent systems', in *International workshop on agent-oriented software engineering*, pp. 214–230, (2004).
- [8] Raul Fervari, Fernando R Velázquez-Quesada, and Yanjing Wang, 'Bisimulations for knowing how logics', *The Review of Symbolic Logic*, **15**(2), 450–486, (2022).
- [9] Yuqian Fu, Jiajun Chai, Yuanheng Zhu, and Dongbin Zhao, 'Lilac: Learning a leader for cooperative reinforcement learning', in *2022 IEEE Conference on Games (CoG)*, pp. 49–55, (2022).
- [10] Scott Fujimoto, David Meger, and Doina Precup, 'Off-policy deep reinforcement learning without exploration', in *International conference on machine learning*, pp. 2052–2062, (2019).
- [11] David Ha, Andrew Dai, and Quoc V Le, 'Hypernetworks', *arXiv preprint arXiv:1609.09106*, (2016).
- [12] Siyi Hu, Chuanlong Xie, Xiaodan Liang, and Xiaojun Chang, 'Policy diagnosis via measuring role diversity in cooperative multi-agent rl', in *International Conference on Machine Learning*, pp. 9041–9071, (2022).
- [13] Shariq Iqbal, Christian A Schroeder De Witt, Bei Peng, Wendelin Böhrer, Shimon Whiteson, and Fei Sha, 'Randomized entity-wise factorization for multi-agent reinforcement learning', in *International Conference on Machine Learning*, pp. 4596–4606, (2021).
- [14] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu, 'Graph convolutional reinforcement learning', *arXiv preprint arXiv:1810.09202*, (2018).
- [15] Jiechuan Jiang and Zongqing Lu, 'Learning attentional communication for multi-agent cooperation', *Annual Conference on Neural Information Processing Systems*, (2018).
- [16] Qize Jiang, Minhao Qin, Shengmin Shi, Weiwei Sun, and Baihua Zheng, 'Multi-agent reinforcement learning for traffic signal control through universal communication method', *International Joint Conference on Artificial Intelligence*, (2022).
- [17] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajkac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al., 'Google research football: A novel reinforcement learning environment', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4501–4510, (2020).
- [18] Kemas M Lhaksana, Yohei Murakami, and Toru Ishida, 'Role modeling for adaptive multiagent systems engineering', in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 2, pp. 287–292, (2013).
- [19] Chuming Li, Jie Liu, Yinmin Zhang, Yuhong Wei, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang, 'Ace: Cooperative multi-agent q-learning with bidirectional action-dependency', *arXiv preprint arXiv:2211.16068*, (2022).
- [20] Kai Liu, Yuyang Zhao, Gang Wang, and Bei Peng, 'Self-attention-based multi-agent continuous control method in cooperative environments', *Information Sciences*, **585**, 454–470, (2022).
- [21] Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira, 'When2com: Multi-agent perception via communication graph grouping', in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 4106–4115, (2020).
- [22] Yuntao Liu, Yuan Li, Xinhai Xu, Donghong Liu, and Yong Dou, 'Rogc: Role-oriented graph convolution based multi-agent reinforcement learning', in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, (2022).
- [23] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch, 'Multi-agent actor-critic for mixed cooperative-competitive environments', *Advances in neural information processing systems*, **30**, (2017).
- [24] Maximilian Nickel and Douwe Kiela, 'Poincaré embeddings for learning hierarchical representations', *Advances in neural information processing systems*, **30**, (2017).
- [25] Yaru Niu, Rohan R Paleja, and Matthew C Gombolay, 'Multi-agent graph-attention communication and teaming', in *AAMAS*, pp. 964–973, (2021).
- [26] James J Odell, H Van Dyke Parunak, and Mitchell Fleischer, 'The role of roles in designing effective agent organizations', in *International Workshop on Software Engineering for Large-Scale Multi-agent Systems*, pp. 27–38, (2002).
- [27] Juan Pavón and Jorge Gómez-Sanz, 'Agent oriented software engineering with ingenias', in *International Central and Eastern European Conference on Multi-Agent Systems*, pp. 394–403, (2003).
- [28] Thomy Phan, Fabian Ritz, Lenz Belzner, Philipp Altmann, Thomas Gabor, and Claudia Linnhoff-Popien, 'Vast: Value function factorization with variable agent sub-teams', *Advances in Neural Information Processing Systems*, **34**, 24018–24032, (2021).
- [29] Henri Poincaré, 'Théorie des groupes fuchsien', *Acta mathematica*, **1**(1), 1–62, (1882).
- [30] J Ruan, Y Du, X Xiong, D Xing, X Li, L Meng, H Zhang, J Wang, and B Xu, 'Gcs: Graph-based coordination strategy for multi-agent reinforcement learning', in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 2, pp. 1128–1136, (2022).
- [31] Jingqing Ruan, Linghui Meng, Xuantang Xiong, Dengpeng Xing, and Bo Xu, 'Learning multi-agent action coordination via electing first-move agent', in *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pp. 624–628, (2022).
- [32] Yu Tian, Xingliang Huang, Ruigang Niu, Hongfeng Yu, Peijin Wang, and Xian Sun, 'Hypertron: Explicit social-temporal hypergraph framework for multi-agent forecasting', (2022).
- [33] Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, et al., 'Jump-start reinforcement learning', *arXiv preprint arXiv:2204.02372*, (2022).
- [34] Aaron Van Den Oord, Oriol Vinyals, et al., 'Neural discrete representation learning', *Advances in neural information processing systems*, **30**, (2017).
- [35] T Wang, T Gupta, B Peng, A Mahajan, S Whiteson, and C Zhang, 'Rode: learning roles to decompose multi-agent tasks', in *Proceedings of the International Conference on Learning Representations*, (2021).
- [36] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang, 'Roma: Multi-agent reinforcement learning with emergent roles', in *International Conference on Machine Learning*, pp. 9876–9886, (2020).
- [37] Zhiwei Xu, Bin Zhang, Dapeng Li, Zeren Zhang, Guangchong Zhou, and Guoliang Fan, 'Consensus learning for cooperative multi-agent reinforcement learning', *Advances in Neural Information Processing Systems*, (2022).
- [38] Mingyu Yang, Jian Zhao, Xunhan Hu, Wengang Zhou, and Houqiang Li, 'Ldsa: Learning dynamic subtask assignment in cooperative multi-agent reinforcement learning', *arXiv preprint arXiv:2205.02561*, (2022).
- [39] Lei Yuan, Chenghe Wang, Jianhao Wang, Fuxiang Zhang, Feng Chen, Cong Guan, Zongzhang Zhang, Chongjie Zhang, and Yang Yu, 'Multi-agent concentrative coordination with decentralized task representation', in *International Joint Conference on Artificial Intelligence*, (2022).
- [40] Chongjie Zhang and Victor Lesser, 'Coordinated multi-agent reinforcement learning in networked distributed pomdps', in *Twenty-Fifth AAAI Conference on Artificial Intelligence*, (2011).
- [41] Xianjie Zhang, Yu Liu, Hangyu Mao, and Chao Yu, 'Common belief multi-agent reinforcement learning based on variational recurrent models', *Neurocomputing*, **513**, 341–350, (2022).