BEYOND SINGLE ROLES: STRUCTURED ROLE DISCOVERY FOR COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Role-based multi-agent reinforcement learning (MARL) learns roles to organize agents into specialized divisions of labor. Effective coordination requires balancing knowledge sharing with behavioral diversity. However, a central challenge is to move beyond current role assignments toward persistent role structures that capture inter-role associations and support reuse across episodes. To address this challenge, we propose SPAR, which constructs a role prototype space using semantic slots and represents each agent through a dual semantic assignment consisting of a primary role and an auxiliary role. Building on this representation, SPAR employs relation-preserving contrastive learning to optimize the geometric structure of the role space and introduces a performance-guided slot memory to reuse high-quality collaboration patterns. Experiments on SMAC and SMACv2 demonstrate that SPAR can learn dynamic yet structured roles and achieves competitive performance across multiple test scenarios.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) investigates how multiple agents learn coordinated decision-making policies under local observations and shared team rewards, with applications spanning unmanned aerial vehicle (UAV) swarms, autonomous driving, traffic control, and real-time strategy games Pham et al. (2018); Zhou et al. (2020); Chu et al. (2019); Vinyals et al. (2019). Centralized training with decentralized execution (CTDE) has become a dominant paradigm in cooperative MARL Foerster et al. (2016); Sunehag et al. (2018); Wang et al. (2020a); Yang et al. (2020). Representative CTDE methods, including multi-agent deep deterministic policy gradient (MADDPG) Lowe et al. (2017), QMIX Rashid et al. (2020), and multi-agent proximal policy optimization (MAPPO) Yu et al. (2022), have advanced cooperative policy learning, with QMIX and MAPPO demonstrating strong performance on the StarCraft Multi-Agent Challenge (SMAC) Samvelyan et al. (2019). Within this paradigm, parameter sharing is widely adopted to improve learning efficiency and scalability, allowing agents to learn from pooled experience while reducing the number of independently trained parameters Christianos et al. (2021); Fu et al. (2022).

However, although parameter sharing improves training efficiency and scalability Christianos et al. (2021); Fu et al. (2022), it can also lead to behavioral homogeneity and insufficient role differentiation, thereby restricting exploratory diversity and fine-grained coordination Li et al. (2021); Jiang & Lu (2021); Hu et al. (2022). This tendency is particularly limiting in complex cooperative tasks, where coordinated pursuit of a shared objective often depends on complementary role specialization. Role-based MARL addresses this issue by learning latent role representations and inducing complementary divisions of labor within a parameter-sharing framework. Along this line, ROMA Wang et al. (2020b) learns emergent roles to promote policy specialization and adaptive sharing among agents. RODE Wang et al. (2021) decomposes tasks through role-specific action spaces, reducing learning complexity and improving scalability.

Beyond role differentiation, another challenge is to connect agents' distinct behaviors with their contributions to team coordination. Contrastive identity-aware learning (CIA) Liu et al. (2023) addresses this challenge by using contrastive learning to distinguish individual contributions within value decomposition. Building on this idea, later work moves beyond distinguishing individual

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

contributions toward explicitly organizing differentiated agent behaviors. SIRD Zeng et al. (2023) formulates role discovery as hierarchical action-space clustering guided by structural information principles.At the group level, GoMARL Zang et al. (2023) combines automatic grouping with hierarchical value decomposition to support intra- and inter-group coordination.Furthermore, ACORM Hu et al. (2024) learns role representations from agent trajectories through contrastive learning and integrates them into attention-based value decomposition, linking behavioral differentiation with cooperative credit assignment.

Despite continued progress in modeling role differentiation and coordination structures, existing approaches are still commonly constrained by an instantaneous single-role assumption Wang et al. (2020b; 2021); Zeng et al. (2023); Hu et al. (2024); Goel et al. (2025). Two limitations remain in how role structures are represented and retained, a mutually exclusive role label captures only the dominant responsibility and obscures complementary or transitional relations among roles. Across episodes of interaction with the environment, role assignments are regenerated from current trajectories rather than retained, preventing effective coordination structures from being accumulated and reused in similar situations .Together, these limitations reveal a gap between discovering role assignments and maintaining reusable role structures with explicit inter-role associations.To bridge this gap, SPAR introduces primary–auxiliary role assignments that represent each agent's dominant and secondary slot associations.Unlike ACORM's single-label contrastive supervision Hu et al. (2024), our objective uses these secondary associations to downweight related cross-role negatives.A performance-guided slot memory further stores historical slot structures and selectively retrieves them for reuse in similar situations.

- We propose slot-semantic role discovery Locatello et al. (2020), which refines role prototypes through assignment-weighted aggregation and recurrent updates, then extracts each agent's primary and auxiliary roles from its two highest slot-assignment scores.
- We develop a primary–auxiliary role-aware contrastive objective Oord et al. (2018), which constructs positive and negative pairs according to primary-role labels and downweights cross-role negatives when either agent's auxiliary role matches the other's primary role.
- We design a performance-guided slot memory Pritzel et al. (2017), which stores slot states with contextual features and episode performance, retrieves candidates through coarse similarity filtering and agent-level matching, and selects the highest-performing candidate for similarity-gated reuse.

2 Related Work

2.1 Role-Based Multi-Agent Reinforcement Learning

ROMA learns continuous role embeddings from agents' current observations and observation-action histories, representing one of the early approaches to emergent role learning in MARL (Wang et al., 2020b). Building on this idea, RODE and SIRD associate different roles with predefined action subspaces to reduce policy search complexity and simplify policy learning (Wang et al., 2021; Zeng et al., 2023). However, such methods typically constrain each role to a relatively fixed set of actions, which may limit the flexibility required for dynamic coordination.

In contrast, LDSA and ALMA enable dynamic role assignment by learning latent subtasks (Yang et al., 2022; Iqbal et al., 2022). Contrastive learning has also been introduced to improve the efficiency of role representation learning (Song et al., 2023; Li et al., 2024). CIA distinguishes agent roles according to their contributions to the team objective (Liu et al., 2023). ACORM clusters agent behavior representations and constructs a contrastive learning objective using pseudo role labels to capture differentiated behavioral patterns (Hu et al., 2024). Methods such as SePS and GoMARL further promote agent heterogeneity and specialization through dynamic grouping (Christianos et al., 2021; Zang et al., 2023). R3DM takes a different perspective by focusing on differences in expected future behaviors and employs intrinsic rewards to encourage distinct roles to develop differentiated future behavioral patterns (Goel et al., 2025).

Unlike these approaches, our method performs a fine-grained decomposition of the role structure through soft assignment over semantic slots, thereby explicitly characterizing each agent in terms of a primary role and an auxiliary role.

2.2 Episodic Memory in Multi-Agent Reinforcement Learning

Episodic memory supports current policy learning and decision-making by storing and retrieving high-value experiences from previous interactions. EMC combines episodic memory with intrinsic rewards, using value prediction errors to encourage agents to explore novel states while leveraging historical experiences to improve cooperative policies (Zheng et al., 2021). EMU, in contrast, learns semantically consistent memory representations and retrieves similar high-value experiences according to the current state, thereby improving the reliability and efficiency of episodic memory retrieval and utilization (Na et al., 2024).

Unlike existing approaches that primarily reuse historical states, value information, or trajectory-level experiences, our method stores the team-level role structures formed during high-performing episodes in SlotBank and retrieves and reuses them under similar coordination contexts.

3 Method

SPAR combines primary–auxiliary role assignments, role-aware contrastive learning, and historical slot reuse within a framework. As illustrated in Figure 1, the upper part depicts environment interaction and value decomposition, while the middle and lower parts detail the role-learning components. After presenting the task formulation and backbone, we describe these components in three subsections. First, slot-semantic role discovery refines slot prototypes from agent embeddings and generates primary and auxiliary assignments, corresponding to the middle-left module. Second, the role-aware contrastive objective constructs positive and negative pairs from primary assignments and adjusts cross-role negative weights using auxiliary assignments. This objective trains the representations in the middle-right module. Finally, performance-guided slot memory stores historical slot structures together with contextual features and episode performance, as shown in the bottom module. Similarity matching identifies candidates, and historical performance selects the entry.

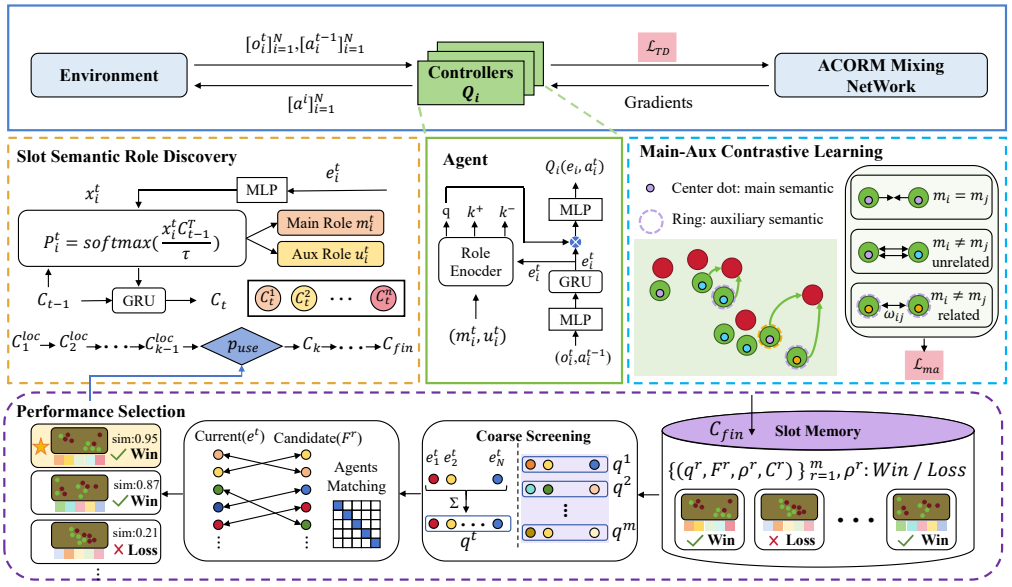


Figure 1: **The SPAR framework based on QMIX.** The upper part shows the overall value-decomposition architecture. The middle part depicts slot-semantic role discovery, the shared individual Q-network, and main-auxiliary role-aware contrastive learning, which preserves role discriminability while retaining weak associations between related roles. The lower part illustrates performance-guided slot memory: historical role structures are retrieved through coarse screening and agent-level matching, selected according to past performance, and reused to guide role discovery in subsequent episodes.

We formulate the multi-agent task as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek & Amato, 2016), defined as

$$\mathcal{G} = \langle I, S, A, P, R, \Omega, O, n, \gamma \rangle, \tag{1}$$

where I is the agent set, S is the global state space, and A is the individual action space. $P(s' \mid s, \mathbf{a})$ specifies the transition probability from state s to s' under joint action $\mathbf{a}$, and $R(s, \mathbf{a})$ is the shared team reward function. $\Omega(s, i)$ maps the global state and agent index to a local observation in the observation space O. $n = |I|$ is the number of agents, and $\gamma \in [0, 1)$ is the discount factor. At each timestep, agent i receives $o_i = \Omega(s, i)$ and selects $a_i \in A$ based on its local action-observation history. After the joint action

$$\mathbf{a} = [a_1, \ldots, a_n]^\top \in A^n \tag{2}$$

is executed, the environment transitions to the next state s' according to the transition probability $P(s'|s, \mathbf{a})$ and provides all agents with a shared team reward

$$r = R(s, \mathbf{a}). \tag{3}$$

Following the centralized training with decentralized execution (CTDE) paradigm, we adopt value decomposition to learn decentralized individual utility functions while estimating a centralized joint action-value function. Specifically, QMIX (Rashid et al., 2020) factorizes the joint action-value function Q_{tot} into a monotonic combination of individual action-value functions:

$$Q_{\text{tot}}(s, \mathbf{a}) = f_{\text{mix}}\left(Q_1(\tau_1, a_1), \ldots, Q_n(\tau_n, a_n), s\right), \tag{4}$$

where f_{mix} denotes the mixing network conditioned on the global state s. $Q_i(\tau_i, a_i)$ denotes the individual utility of agent i, conditioned on its local action-observation history τ_i and action a_i.

Building on this backbone, SPAR extends ACORM's role-conditioned QMIX framework (Hu et al., 2024) with structured role modeling and cross-episode role-structure reuse. These extensions are designed to capture inter-role relationships within an episode while preserving effective coordination structures across episodes. We detail the three components below.

3.1 Slot-Semantic Role Discovery

SPAR first reformulates the role discovery mechanism in ACORM. ACORM generates hard role pseudo-labels by clustering agent embeddings and uses these labels to construct positive and negative pairs for contrastive learning (Hu et al., 2024). Although this mechanism can effectively distinguish agents exhibiting different behavioral patterns, each role is represented by a single discrete label, making it difficult to capture secondary responsibilities that an agent may assume during coordination or potential transitions between roles. To address this limitation, SPAR formulates role discovery as a structured assignment process induced by semantic slots, such that each agent is characterized not only by a primary role but also by an auxiliary role.

Specifically, given the behavioral representation e_i^t of agent i at timestep t, the model first projects it into the slot-semantic space to obtain a normalized representation x_i^t. Meanwhile, a set of recurrently refined slot states

$$C_{t-1} = \left\{c_k^{t-1}\right\}_{k=1}^{K} \tag{5}$$

is maintained, where each slot represents a latent role prototype. The semantic affinity between agents and slots is modeled through a temperature-scaled soft assignment:

$$P_{ik}^t = \frac{\exp\left((x_i^t)^\top c_k^{t-1}/\tau_s\right)}{\sum_{r=1}^{K} \exp\left((x_i^t)^\top c_r^{t-1}/\tau_s\right)}, \tag{6}$$

where P_{ik}^t denotes the assignment strength of agent i to the k-th slot, and τ_s controls the smoothness of the role assignment distribution. Each slot then aggregates the representations of its associated agents through assignment-weighted pooling and is updated via recurrent refinement (Locatello et al., 2020):

$$c_k^t = \text{Refine}_\psi\left(c_k^{t-1}, \frac{\sum_{i=1}^{n} P_{ik}^t x_i^t}{\sum_{i=1}^{n} P_{ik}^t + \epsilon}\right). \tag{7}$$

This recurrent update allows the slots to evolve beyond static clustering centers and instead serve as dynamic role prototypes that adapt to the evolving team behaviors within an episode. Based on the resulting assignment distribution, SPAR identifies the primary and auxiliary roles of each agent as

$$m_i^t = \mathrm{Top1}(P_i^t), \qquad u_i^t = \mathrm{Top2}(P_i^t). \tag{8}$$

Here, m_i^t captures the agent's most salient behavioral role at the current timestep, whereas u_i^t characterizes its latent association with a secondary role. In this way, SPAR extends the single hard role label used in ACORM into a primary–auxiliary role structure, preserving role discriminability while providing a finer-grained representation of collaborative semantics.

3.2 Main-Auxiliary Role-Aware Contrastive Objective

Following ACORM's information-theoretic motivation for role representation learning (Hu et al., 2024), SPAR constructs positive pairs using shared primary-role labels and negative pairs using different primary-role labels (Khosla et al., 2020). We further incorporate auxiliary-role associations to distinguish potentially related negatives. The resulting weighted multi-positive objective admits the candidate-conditioned information bound established below.

However, in cooperative multi-agent tasks, different primary roles do not necessarily imply semantic irrelevance. For example, one agent may be responsible for drawing enemy fire while another focuses on concentrated damage output. Although their primary roles differ, their contributions to team performance can be closely coupled. Uniformly repelling such pairs may therefore obscure relationships relevant to coordination (Chuang et al., 2020). This motivates distinguishing potentially related cross-role pairs from other negatives, rather than treating all pairs with different primary-role labels identically.

To address this issue, SPAR retains samples with the same primary role as positives and those with different primary roles as negatives, while using auxiliary-role information to modulate the contribution of potentially related negatives. Let z_i^t denote the query role representation of agent i, and let $\bar{z}_j^t$ denote the target role representation of agent j. Their bilinear similarity is defined as

$$s_{ij}^t = (z_i^t)^\top W \bar{z}_j^t. \tag{9}$$

For two agents i and j with different primary roles, if

$$u_i^t = m_j^t \qquad \text{or} \qquad u_j^t = m_i^t, \tag{10}$$

then the auxiliary role of one agent coincides with the primary role of the other. SPAR treats such pairs as potentially related negatives and downweights their contribution to the contrastive denominator:

$$\omega_{ij}^t = 1 - \alpha\, \mathbb{I}\left[u_i^t = m_j^t \text{ or } u_j^t = m_i^t\right] \eta_{ij}^t, \tag{11}$$

where α controls the attenuation of related negatives, and η_{ij}^t measures the strength of the primary–auxiliary association. For the analysis below, assume $0 \le \eta_{ij}^t \le \eta_{\max}$ and $0 \le \alpha\eta_{\max} < 1$, yielding the uniform weight bound $0 < \underline{\omega} = 1 - \alpha\eta_{\max} \le \omega_{ij}^t \le 1$.

Let $\mathcal{V}$ denote the set of valid anchor–time pairs, each having at least one positive candidate. The primary–auxiliary role-aware contrastive objective is defined as

$$\mathcal{L}_{\mathrm{ma}} = -\frac{1}{|\mathcal{V}|} \sum_{(i,t)\in\mathcal{V}} \log \frac{\sum_{j:\, m_j^t = m_i^t} \exp(s_{ij}^t)}{\sum_{j:\, m_j^t = m_i^t} \exp(s_{ij}^t) + \sum_{j:\, m_j^t \neq m_i^t} \omega_{ij}^t \exp(s_{ij}^t)}. \tag{12}$$

Theorem 1 (Conditional information bound). *Let $\mathcal{D} = \{(\bar{Z}_j, M_j)\}_{j=1}^n$ be the labeled candidate set, and let (M, Z) denote the primary role and representation of a valid anchor. Suppose each anchor has at least one positive candidate and all negative weights satisfy $0 < \underline{\omega} \le \omega_j \le 1$. Let $\overline{\mathcal{L}}_{\mathrm{ma}}$ denote the expected per-anchor loss in Eq. equation 12, under the same joint sampling distribution of anchors and candidates. Then*

$$I(M; Z \mid \mathcal{D}) \ge H(M \mid \mathcal{D}) - \overline{\mathcal{L}}_{\mathrm{ma}} + \log \underline{\omega}. \tag{13}$$

Proof Sketch. We first establish Lemma A.1, which bounds conditional mutual information using the unweighted contrastive loss through non-negativity of conditional KL divergence. We then use the negative-weight bounds to show that the unweighted expected loss exceeds the weighted loss by at most $-\log \underline{\omega}$. Combining this loss comparison with Lemma A.1 yields the stated bound. The full proof is provided in Appendix A.1. □

3.3 PERFORMANCE-GUIDED SLOT MEMORY

The preceding two modules perform structured role discovery and role representation optimization primarily within individual episodes. In cooperative multi-agent tasks, however, similar situations may recur across episodes, making previously discovered role organizations potentially reusable. Reconstructing slot structures solely from the current episode's agent embeddings does not explicitly exploit these historical structures and may result in repeated exploration of similar role organizations.

To address this issue, SPAR introduces a performance-guided slot memory that stores high-performing historical slot structures and enables their controlled reuse in subsequent episodes. The memory bank is defined as

$$\mathcal{B} = \{(q^r, F^r, \rho^r, C^r)\}_{r=1}^{B}, \tag{14}$$

where q^r is a situation key obtained by permutation-invariant aggregation of historical agent features (Zaheer et al., 2017), F^r contains the corresponding agent-level features, ρ^r records the final performance of that episode, and C^r denotes the stored slot structure. Unlike a conventional replay buffer that stores individual transitions, this memory stores episode-level role structures together with their associated situations and historical performance.

For the current episode, SPAR constructs a situation key from early-stage agent features using the same aggregation and coarsely filters memory entries by key similarity. It then refines candidate similarity through Hungarian matching between agent-level features (Kuhn, 1955). For row-normalized feature matrices $F = (f_i)_{i=1}^n$ and $F^r = (f_j^r)_{j=1}^n$, the matching score is

$$h(F, F^r) = \frac{1}{n} \max_{\pi \in \mathfrak{S}_n} \sum_{i=1}^{n} f_i^\top f_{\pi(i)}^r, \tag{15}$$

where $\mathfrak{S}_n$ denotes the set of permutations of n agents. This score compares agent-feature sets independently of their ordering; it does not reorder the stored slots. Among the retained candidates, the model selects the entry with the highest historical performance:

$$r^* = \arg \max_{r \in \mathcal{C}(q,F)} \rho^r, \tag{16}$$

where $\mathcal{C}(q, F)$ is the candidate set retained by similarity-based filtering and matching. Historical performance ranks the retained candidates, whereas the following gate controls whether the selected structure is reused. Ties in candidate filtering and selection are resolved using a fixed rule independent of agent ordering.

To control historical reuse, SPAR combines the selected candidate's absolute similarity with its relative similarity margin to determine the reuse probability:

$$p_{\text{reuse}} = \text{Sim}\left(q, F; q^{r^*}, F^{r^*}\right) \cdot \sigma\left(\tau_m \Delta_{r^*}\right), \tag{17}$$

where $\text{Sim}(q, F; q^{r^*}, F^{r^*}) = \text{clip}_{[0,1]} h(F, F^{r^*})$. σ denotes the sigmoid function, and $\tau_m > 0$ controls sensitivity to the relative similarity margin. Δ_{r^*} is the selected matching score minus the mean score of the remaining candidates in $\mathcal{C}(q, F)$. When only one candidate is retained, Δ_{r^*} is set to zero.

The final slot state is given by

$$C^{\text{out}} = \begin{cases} C^{r^*}, & \text{with probability } p_{\text{reuse}}, \\ C^{\text{loc}}, & \text{otherwise}, \end{cases} \tag{18}$$

where C^{loc} denotes the locally generated slots. If no memory candidate is available, the model retains C^{loc}.

Theorem 2 (Permutation consistency). *Let $P(X, C_0)$ and $\Phi(X, C_0)$ denote the assignment matrix and refined slots obtained from agent features X and fixed initial slots C_0, respectively. Suppose slot discovery uses shared agent-wise mappings and symmetric aggregation, and memory retrieval uses permutation-invariant pooled keys and exact one-to-one matching. For any agent permutation matrices Π and Γ,*

$$\begin{aligned} P(\Pi X, C_0) &= \Pi P(X, C_0), \\ \Phi(\Pi X, C_0) &= \Phi(X, C_0), \\ h(\Pi F, \Gamma F^r) &= h(F, F^r). \end{aligned} \tag{19}$$

With the memory entries and their historical performance held fixed, and tie-breaking independent of agent ordering, the selected memory entry and reuse probability remain unchanged under these permutations, and C^{out} has the same distribution.

Proof Sketch. We first show that agent permutations reorder assignment rows but leave assignment-weighted aggregation unchanged. Induction over refinement steps then establishes slot invariance. Next, we prove matching-score invariance by showing that permutations bijectively reindex the feasible one-to-one matchings. With fixed memory contents and order-independent tie-breaking, these invariances preserve candidate selection and the reuse probability, yielding the same output distribution. The full proof is provided in Appendix A.2. □

4 EXPERIMENTS

We evaluate SPAR on SMAC and SMACv2, examining overall performance, learning dynamics, primary–auxiliary role assignments, and the contributions of individual components.

Environments. We evaluate SPAR on SMAC and SMACv2 (Samvelyan et al., 2019; Ellis et al., 2023). The six SMAC scenarios are 5m_vs_6m, MMM2, 3s5z_vs_3s6z, 27m_vs_30m, 6h_vs_8z, and corridor. SMACv2 introduces variation in unit types, starting positions, and team configurations. All experiments use SC2 version 4.10. **Baselines.** We compare SPAR with ACORM (Hu et al., 2024), QMIX (Rashid et al., 2020), CIA (Liu et al., 2023), GoMARL (Zang et al., 2023), CDS (Li et al., 2021), and EMC (Zheng et al., 2021).

4.1 PERFORMANCE COMPARISON

We compare final test win rates on SMAC and SMACv2 and learning curves on the six SMAC scenarios.

Table 1: Final test win rate (%) on SMAC and SMACv2.

Benchmark	Map	**SPAR (Ours)**	ACORM	CIA	QMIX	CDS	GoMARL	EMC
SMAC	3s5z_vs_3s6z	**77.5 ± 16.3**	49.9 ± 20.3	30.1 ± 38.2	0.4 ± 0.6	42.0 ± 23.4	7.0 ± 4.8	46.2 ± 31.3
	corridor	**83.5 ± 6.2**	65.3 ± 38.5	4.7 ± 8.3	0.0 ± 0.0	65.5 ± 10.1	49.2 ± 41.8	63.4 ± 35.7
	6h_vs_8z	**18.4 ± 21.1**	2.9 ± 2.8	11.6 ± 25.6	0.0 ± 0.0	0.4 ± 0.3	1.0 ± 0.9	0.0 ± 0.0
	27m_vs_30m	**77.1 ± 7.2**	68.9 ± 3.5	**74.8 ± 10.8**	20.1 ± 11.5	0.0 ± 0.0†	26.3 ± 6.5	7.5 ± 13.1
	5m_vs_6m	65.9 ± 10.7	56.5 ± 13.0	**70.1 ± 5.1**	26.4 ± 19.1	52.1 ± 15.6	58.5 ± 8.1	48.0 ± 10.9
	MMM2	**85.5 ± 3.6**	**86.2 ± 7.9**	12.6 ± 18.4	41.6 ± 17.1	73.5 ± 19.1	38.8 ± 13.7	3.1 ± 6.3
SMACv2	Protoss 5v5	**57.5 ± 5.4**	**56.4 ± 2.6**	46.5 ± 10.1	38.8 ± 2.4	46.2 ± 6.7	51.4 ± 6.4	48.9 ± 17.2
	Terran_5_vs_5	**63.9 ± 1.4**	**61.6 ± 3.3**	57.0 ± 4.8	40.4 ± 5.2	56.6 ± 5.4	60.5 ± 4.9	40.6 ± 22.2
	Zerg_5_vs_5	**44.9 ± 1.9**	36.6 ± 3.4	38.6 ± 5.3	21.5 ± 3.8	24.9 ± 6.0	**43.9 ± 6.6**	20.1 ± 4.3
	Protoss_10_vs_11	17.4 ± 3.7	13.9 ± 6.8	16.2 ± 4.1	5.4 ± 1.9	1.4 ± 1.2	**18.8 ± 4.9**	3.0 ± 6.7
	Terran_10_vs_11	25.8 ± 8.4	21.6 ± 5.0	22.0 ± 3.4	2.8 ± 2.9	5.6 ± 1.8	**35.8 ± 4.5**	0.0 ± 0.0
	Zerg_10_vs_11	23.0 ± 1.9	14.3 ± 7.1	**29.4 ± 9.2**	8.1 ± 2.8	1.2 ± 0.9	**28.9 ± 3.5**	2.6 ± 3.9
Average		**53.4**	44.5	34.5	17.1	30.8	35.0	23.6

Values are test win rate (%). For each seed, we average its final five recorded evaluations within the task budget, then report mean ± sample standard deviation across seeds. All entries use five seeds except †, which uses four. Bold indicates results within 5% of the best mean in the row. The Average row averages the 12 per-task means.

As shown in Table 1, SPAR achieves the highest average test win rate of 53.4% across the 12 evaluated tasks, outperforming ACORM by 8.9 percentage points. It obtains the best mean performance on 7 tasks and performs within 5% of the best mean on 8 tasks, while outperforming ACORM on

11 out of 12 scenarios. The gains are particularly evident on challenging SMAC scenarios such as 3s5z_vs_3s6z, corridor, and 6h_vs_8z. On SMACv2, SPAR achieves the best performance on all three 5_vs_5 scenarios and consistently improves over ACORM on all three 10_vs_11 scenarios, despite GoMARL and CIA remaining stronger on several large-scale tasks. Beyond final performance, we further examine whether these improvements are also reflected in the learning process on the six SMAC scenarios.

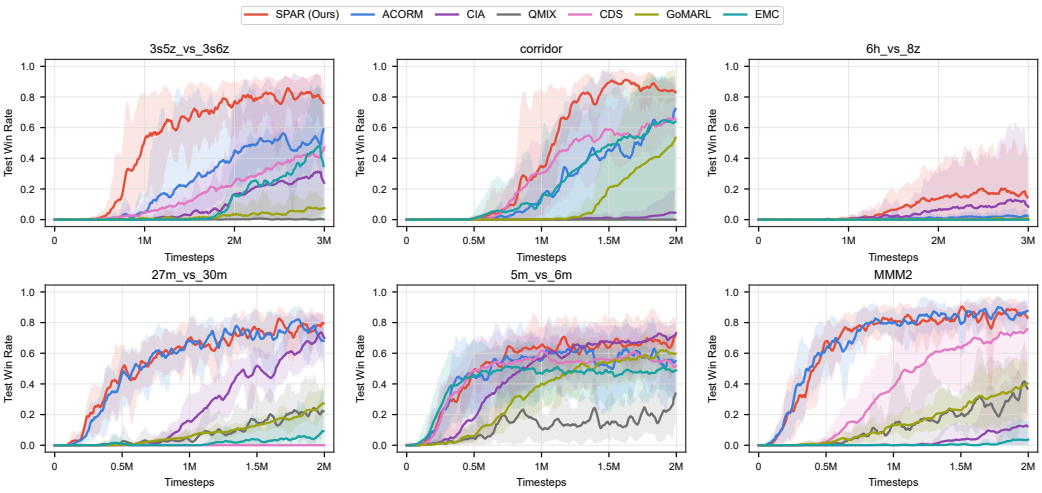


Figure 2: Learning curves on six SMAC scenarios.

Figure 2 further shows that the performance gains of SPAR are accompanied by favorable learning dynamics on several challenging scenarios. On 3s5z_vs_3s6z and corridor, SPAR learns effective coordination policies relatively early and maintains higher test win rates during the middle and later stages of training. On 6h_vs_8z, where most baselines remain close to zero for much of the training process, SPAR gradually learns a more effective cooperative policy. Meanwhile, its learning behavior on 27m_vs_30m and MMM2 remains comparable to the strongest baselines, indicating that the additional role structure does not compromise performance on scenarios where ACORM is already competitive. Together with the consistent improvements observed on SMACv2, these results demonstrate that SPAR provides a robust improvement over the underlying ACORM framework across diverse cooperative settings. The following experiments further investigate the learned role structures and the contribution of individual components.

4.2 Qualitative Analysis

Figure 3 provides a qualitative view of how SPAR reorganizes primary and auxiliary roles within a representative trajectory on the 3s5z_vs_3s6z scenario.

From the initial formation to late-stage coordination, primary assignments remain comparatively more stable, whereas auxiliary assignments adapt more frequently to the evolving coordination context, with 16 auxiliary-role switches versus 12 primary-role switches across the displayed transitions. The transition matrix further shows that these changes are agent-specific: some agents retain their primary responsibility while updating only their auxiliary association, whereas others revise both assignments alongside formation changes and agent elimination.

This pattern is consistent with the intended dual-semantic representation, in which the primary role captures the dominant responsibility and the auxiliary role preserves complementary or transitional relations that a mutually exclusive single-role label would discard. As this is a single trajectory, the visualization is illustrative rather than statistical evidence; nevertheless, it provides a concrete example of the dynamic yet structured within-episode role organization targeted by SPAR.

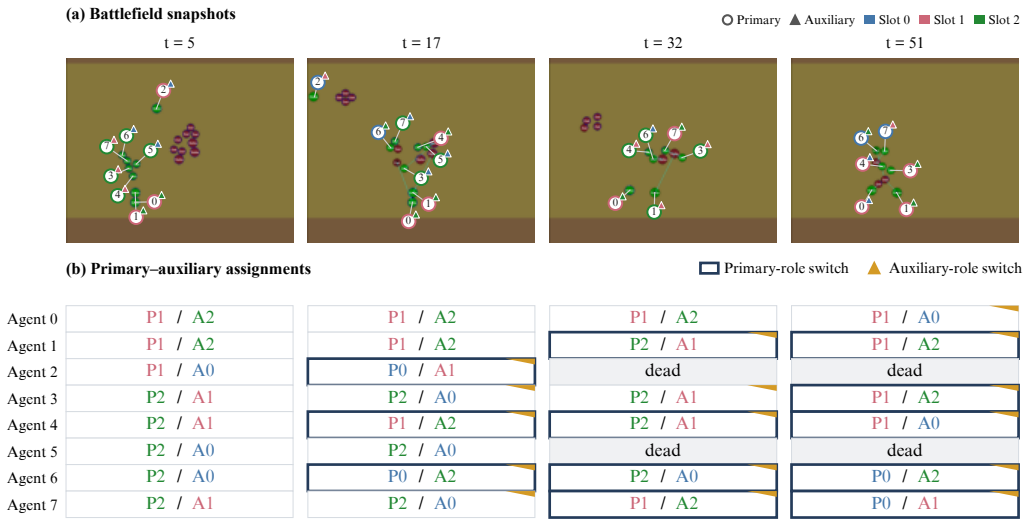


Figure 3: Primary–auxiliary roles in one 3s5z_vs_3s6z evaluation trajectory. (a) Battlefield snapshots. (b) Primary (P) and auxiliary (A) slot assignments; dead denotes eliminated agents. Switch counts include only agents alive at both adjacent displayed stages.

4.3 Ablation

We compare SPAR with variants that disable auxiliary-aware negative weakening (w/o AuxWeak; $\alpha = 0$), slot-memory reuse (w/o SlotBank), or both (w/o Both), while retaining slot-based assignments. Figure 4 shows higher late-training mean win rates for the full method than for these variants, although returns are similar to w/o AuxWeak. These results suggest that combining auxiliary-aware negative weakening with slot-memory reuse improves performance on this scenario.

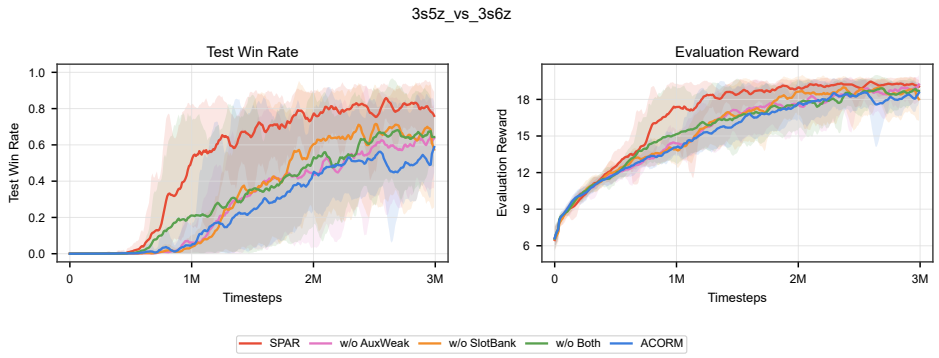


Figure 4: Ablations on 3s5z_vs_3s6z. Curves show smoothed means over five seeds; shaded bands show smoothed across-seed minima and maxima.

5 Conclusion

We propose SPAR for structured role learning in cooperative MARL. Semantic slots encode primary–auxiliary assignments, a role-aware contrastive objective preserves related-role affinities, and performance-guided memory reuses historical slot structures. On SMAC and SMACv2, SPAR achieves higher mean win rates than ACORM on 11 of 12 tasks, although gains vary across scenarios. The predefined slot count remains a limitation, motivating adaptive role discovery (Fan et al., 2024) and further investigation of memory retrieval reliability.

AI USE STATEMENT

Generative AI tools were used to assist with translation, language editing, manuscript organization, LaTeX formatting, and improving the clarity and readability of technical descriptions. They were also used to assist with checking the consistency of terminology, notation, and presentation across different parts of the manuscript. In these uses, the tools served as writing and editing aids rather than as sources of scientific evidence or independent research judgment.

The authors reviewed and verified all AI-assisted edits, compared the resulting text with the underlying method, implementation, experimental records, and cited sources where appropriate, and revised the manuscript as necessary. References, numerical results, and technical statements were not accepted solely on the basis of AI-generated suggestions, but were checked against the corresponding source materials and experimental records. No AI-generated text was incorporated into the manuscript without author review and revision.

Generative AI was not used to generate research ideas, conduct experiments, select reported results, or determine scientific conclusions. The design of the method, experimental setup, interpretation of the results, and final scientific judgments remained the responsibility of the authors. The authors take full responsibility for the final text, claims, and presentation of the paper.

ETHICS STATEMENT

This work studies cooperative multi-agent reinforcement learning in the simulated SMAC and SMACv2 benchmarks. The reported experiments do not involve human participants, personal data, or interaction with real-world physical systems. These controlled environments enable the study of coordination, role organization, and learning dynamics without directly affecting people or deployed autonomous agents. Nevertheless, performance in simulation does not establish the safety, reliability, or robustness of real-world deployment. Learned coordination strategies may behave unexpectedly under distribution shifts, incomplete or noisy observations, changes in team composition, or environmental conditions that differ substantially from those represented in the benchmarks. Such behaviors may be particularly important when coordination policies are transferred to systems with physical consequences or limited opportunities for human intervention. Applications involving physical agents would therefore require additional evaluation of robustness, operational constraints, failure modes, monitoring mechanisms, and unintended collective behaviors before deployment. The present results should be interpreted as evidence about benchmark performance, structured role learning, and coordination behavior in controlled simulation, rather than as validation for deployment in safety-critical or real-world decision-making settings.

REPRODUCIBILITY STATEMENT

The main paper describes the primary–auxiliary role-assignment mechanism, role-aware contrastive objective, and slot-memory retrieval and reuse procedure, together with the QMIX-based value-decomposition backbone used in our implementation. The corresponding equations define the main assignment, weighting, matching, and reuse operations used by the method. The theoretical results state their assumptions, and the appendices provide supporting derivations and proofs. For the QMIX-based implementation, the appendices further document the main network components, representation dimensions, training settings, optimization parameters, slot-memory hyperparameters, and algorithmic pseudocode. These descriptions specify how role learning, contrastive optimization, and historical slot reuse are integrated with value-decomposition training and decentralized action selection.

The experimental section identifies the benchmark scenarios, baseline methods, and evaluation protocol, while the accompanying tables and figure captions report the number of random seeds, aggregation procedures, evaluation metrics, and the meaning of the displayed variability. Where qualitative results are presented, the corresponding captions and text describe how the displayed trajectories and role transitions are interpreted. The ablation descriptions identify the components disabled in each variant so that the contribution of the main design choices can be examined separately. Together, these materials provide the methodological, implementation, and reporting details needed to guide reproduction, facilitate comparison with the reported results, and clarify how the presented experimental conclusions were obtained.

REFERENCES

Filippos Christianos, Georgios Papoudakis, Arrasy Rahman, and Stefano V. Albrecht. Scaling multi-agent reinforcement learning with selective parameter sharing. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1989–1998. PMLR, 2021. URL http://proceedings.mlr.press/v139/christianos21a.html.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3): 1086–1095, 2019.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/63c3ddcc7b23daa1e42dc41f9a44a873-Abstract.html.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/764c18ad230f9e7bf6a77ffc2312c55e-Abstract-Datasets_and_Benchmarks.html.

Ke Fan, Zechen Bai, Tianjun Xiao, Tong He, Max Horn, Yanwei Fu, Francesco Locatello, and Zheng Zhang. Adaptive Slot Attention: Object Discovery with Dynamic Slot Number. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 23062–23071. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02176. URL https://doi.org/10.1109/CVPR52733.2024.02176.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2137–2145, 2016.

Wei Fu, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. Revisiting some common practices in cooperative multi-agent reinforcement learning. In *Proceedings of International Conference on Machine Learning*, pp. 6863–6877, 2022.

Harsh Goel, Mohammad Omama, Behdad Chalaki, Vaishnav Tadiparthi, Ehsan Moradi-Pari, and Sandeep P. Chinchali. R3DM: enabling role discovery and diversity through dynamics models in multi-agent reinforcement learning. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Siyi Hu, Chuanlong Xie, Xiaodan Liang, and Xiaojun Chang. Policy diagnosis via measuring role diversity in cooperative multi-agent RL. In *Proceedings of International Conference on Machine Learning*, pp. 9041–9071, 2022.

Zican Hu, Zongzhang Zhang, Huaxiong Li, Chunlin Chen, Hongyu Ding, and Zhi Wang. Attention-guided contrastive role representations for multi-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/pdf/49724378efbbff7628c17117b3678dbca86ca748.pdf.

Shariq Iqbal, Robby Costales, and Fei Sha. ALMA: Hierarchical learning for composite multi-agent tasks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7155–7166, 2022.

Jiechuan Jiang and Zongqing Lu. The emergence of individuality. In *International Conference on Machine Learning*, pp. 4992–5001. PMLR, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference*

on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html.

Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109. URL https://doi.org/10.1002/nav.3800020109.

Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021.

Tianxu Li, Kun Zhu, Juan Li, and Yang Zhang. Learning Distinguishable Trajectory Representation with Contrastive Loss. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/76227feb18ea0ee40bd15cf02c33e18e-Abstract-Conference.html.

Shunyu Liu, Yihe Zhou, Jie Song, Tongya Zheng, Kaixuan Chen, Tongtian Zhu, Zunlei Feng, and Mingli Song. Contrastive identity-aware learning for multi-agent value decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11595–11603, 2023.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6379–6390, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

Hyungho Na, Yunkyeong Seo, and Il-Chul Moon. Efficient Episodic Memory Utilization of Cooperative Multi-Agent Reinforcement Learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=LjivA1SLZ6.

Frans A Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*, volume 1. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Huy Xuan Pham, Hung Manh La, David Feil-Seifer, and Luan Van Nguyen. Cooperative and distributed reinforcement learning of drones for field coverage. *CoRR*, abs/1803.07250, 2018.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2827–2836. PMLR, 2017.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Haolin Song, Mingxiao Feng, Wengang Zhou, and Houqiang Li. MA2CL: Masked Attentive Contrastive Learning for Multi-Agent Reinforcement Learning. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 4226–4234, 2023.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 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)*, pp. 2085–2087, 2018.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020a.

Tonghan Wang, Heng Dong, Victor R. Lesser, and Chongjie Zhang. ROMA: multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9876–9886. PMLR, 2020b. URL http://proceedings.mlr.press/v119/wang20f.html.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. RODE: learning roles to decompose multi-agent tasks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=TTUVg6vkNjK.

Mingyu Yang, Jian Zhao, Xunhan Hu, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. LDSA: Learning dynamic subtask assignment in cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1698–1710, 2022.

Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, pp. 24611–24624, 2022.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3391–3401, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Automatic grouping for efficient cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=CGj72TyGJy.

Xianghua Zeng, Hao Peng, and Angsheng Li. Effective and stable role-based multi-agent collaboration by structural information principles. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11772–11780, 2023.

Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. In *Advances in Neural Information Processing Systems*, volume 34, pp. 3757–3769, 2021.

Ming Zhou, Jun Luo, Julian Villela, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, Aurora Chongxi Huang, Ying Wen, Kimia Hassanzadeh, Daniel Graves, Dong Chen, Zhengbang Zhu, Nhat M. Nguyen, Mohamed Elsayed, Kun Shao, Sanjeevan Ahilan, Baokuan Zhang, Jiannan Wu, Zhengang Fu, Kasra Rezaee, Peyman Yadmellat, Mohsen Rohani, Nicolas Perez Nieves, Yihan Ni, Seyedershad Banijamali, Alexander I. Cowen-Rivers, Zheng Tian, Daniel Palenicek, Haitham Bou-Ammar, Hongbo

Zhang, Wulong Liu, Jianye Hao, and Jun Wang. SMARTS: scalable multi-agent reinforcement learning training school for autonomous driving. *CoRR*, abs/2010.09776, 2020. URL https://arxiv.org/abs/2010.09776.

A Theoretical Analysis

A.1 Proof of Theorem 1

All expectations below follow the joint sampling distribution of valid anchors and their labeled candidate sets $\mathcal{D} = \{(\bar{Z}_j, M_j)\}_{j=1}^{n}$, with model parameters fixed.

Lemma A.1. *Define*

$$q_W(m \mid z, \mathcal{D}) = \frac{\sum_{j:M_j=m} \exp(z^\top W \bar{Z}_j)}{\sum_j \exp(z^\top W \bar{Z}_j)}. \tag{20}$$

If each valid anchor has a positive candidate, then $\mathcal{L}_0 = \mathbb{E}[-\log q_W(M \mid Z, \mathcal{D})]$ *satisfies*

$$I(M; Z \mid \mathcal{D}) \geq H(M \mid \mathcal{D}) - \mathcal{L}_0. \tag{21}$$

Proof. Since q_W is a normalized role distribution, non-negativity of conditional KL divergence gives

$$\begin{aligned} \mathcal{L}_0 &= H(M \mid Z, \mathcal{D}) + \mathbb{E}_{Z,\mathcal{D}} D_{\mathrm{KL}}\big(p(\cdot \mid Z, \mathcal{D}) \| q_W(\cdot \mid Z, \mathcal{D})\big) \\ &\geq H(M \mid Z, \mathcal{D}). \end{aligned} \tag{22}$$

The result follows from $I(M; Z \mid \mathcal{D}) = H(M \mid \mathcal{D}) - H(M \mid Z, \mathcal{D})$. □

Proof of Theorem 1. For a valid anchor, let $s_j = Z^\top W \bar{Z}_j$ and write

$$A = \sum_{j:M_j=M} e^{s_j}, \qquad B = \sum_{j:M_j \neq M} e^{s_j}, \qquad B_\omega = \sum_{j:M_j \neq M} \omega_j e^{s_j}. \tag{23}$$

Since $A > 0$ and $0 < \underline{\omega} \leq \omega_j \leq 1$,

$$\underline{\omega}(A + B) \leq A + B_\omega \leq A + B. \tag{24}$$

Thus, for $\ell_0 = -\log[A/(A+B)]$ and $\ell_{\mathrm{ma}} = -\log[A/(A+B_\omega)]$,

$$0 \leq \ell_0 - \ell_{\mathrm{ma}} = \log \frac{A+B}{A+B_\omega} \leq -\log \underline{\omega}. \tag{25}$$

Taking expectations and applying Lemma A.1 yields

$$\begin{aligned} I(M; Z \mid \mathcal{D}) &\geq H(M \mid \mathcal{D}) - \mathcal{L}_0 \\ &\geq H(M \mid \mathcal{D}) - \overline{\mathcal{L}}_{\mathrm{ma}} + \log \underline{\omega}. \end{aligned} \tag{26}$$

This completes the proof. □

A.2 Proof of Theorem 2

Proof. We hold model parameters, initial slots, and stored memory entries fixed, and permute only agent-feature rows.

First, shared agent-wise mappings commute with row permutations. For projected features X and any fixed slot state C, row-wise softmax therefore gives

$$P' = \mathrm{softmax}_{\mathrm{row}}\left(\Pi X C^\top / \tau_s\right) = \Pi P. \tag{27}$$

The assignment-weighted feature sums and slot masses satisfy

$$(P')^\top (\Pi X) = P^\top X, \qquad (P')^\top \mathbf{1}_n = P^\top \mathbf{1}_n. \tag{28}$$

Hence each slot receives the same aggregated feature and undergoes the same refinement update. Induction from the fixed initial slots C_0 yields

$$P(\Pi X, C_0) = \Pi P(X, C_0), \qquad \Phi(\Pi X, C_0) = \Phi(X, C_0). \tag{29}$$

Next, let $S = F(F^r)^\top$ and let $\mathcal{P}_n$ be the set of permutation matrices. The matching score satisfies

$$\begin{aligned} h(\Pi F, \Gamma F^r) &= \frac{1}{n} \max_{Q \in \mathcal{P}_n} \langle Q, \Pi S \Gamma^\top \rangle_F \\ &= \frac{1}{n} \max_{Q \in \mathcal{P}_n} \langle \Pi^\top Q \Gamma, S \rangle_F \\ &= \frac{1}{n} \max_{R \in \mathcal{P}_n} \langle R, S \rangle_F = h(F, F^r), \end{aligned} \tag{30}$$

because $Q \mapsto \Pi^\top Q \Gamma$ is a bijection on $\mathcal{P}_n$.

Finally, the pooled keys and all matching scores remain unchanged. With tie-breaking independent of agent ordering, the retained candidates, selected entry, and reuse probability are therefore unchanged. Together with slot invariance, this preserves the conditional output distribution

$$\mathcal{L}(C^{\text{out}}) = p_{\text{reuse}} \delta_{C^{r^*}} + (1 - p_{\text{reuse}}) \delta_{C^{\text{loc}}}, \tag{31}$$

where δ_C is the point mass at C. If no candidate is available, the unchanged local slots are returned. This completes the proof. □

B Implementation Details of QMIX-Based SPAR

Algorithm 1 summarizes the training procedure of QMIX-based SPAR.

Network architecture. SPAR follows the QMIX backbone of ACORM (Hu et al., 2024). The trajectory encoder concatenates the local observation and previous one-hot action, applies a fully connected layer with ReLU, and uses a 128-dimensional GRU followed by a linear output layer. The role encoder maps this embedding to a 64-dimensional vector through a linear layer and sigmoid activation. The shared individual Q-network receives the local observation, previous action, and role representation, and uses a fully connected ReLU layer, a 64-dimensional GRU, and a linear action-value head. For centralized mixing, the state GRU produces n query vectors of dimension 64, which attend to the role representations. The flattened attention output is concatenated with the global state to condition the hypernetworks. The mixing network has a 32-dimensional hidden layer with ELU activation and nonnegative weights generated by two-layer hypernetworks with 64 hidden units. Table 2 summarizes the network configuration.

Table 2: Network configurations of QMIX-based SPAR.

Parameter	Value	Parameter	Value
Trajectory embedding dimension	128	Attention projection dimension	128
Trajectory GRU hidden dimension	128	Attention output dimension	64
Role embedding dimension	64	Attention heads	4
Slot dimension	64	Mixing hidden dimension	32
Slot GRU hidden dimension	64	Mixing activation	ELU
Q-network GRU hidden dimension	64	Hypernetwork hidden dimension	64
State GRU hidden dimension	$64n$	Hypernetwork linear layers	2
State query dimension	64	Role encoder activation	Sigmoid

Slot assignments and contrastive weights. The slot module projects the trajectory embeddings to 64 dimensions, applies layer normalization and ℓ_2 normalization, and initializes each sampled sequence from normalized, orthogonally initialized slot prototypes. It performs one assignment-weighted GRU refinement per step with three slots and assignment temperature $\tau_s = 1$. The primary and auxiliary labels are taken from the two highest scores after refinement. In the soft-weight mode, the association strength in the main-text objective is implemented as

$$g_i^t = \frac{P_{i,u_i^t}^t}{P_{i,m_i^t}^t + P_{i,u_i^t}^t + 10^{-8}}, \qquad \eta_{ij}^t = \max(g_i^t, g_j^t). \tag{32}$$

The gates are detached when constructing negative weights, and $\alpha = 1$. Positive keys include the anchor's own momentum key; candidate pairs are formed within the same sampled episode and timestep. The loss masks padded timesteps, averages over valid anchors at each timestep, and then averages over timesteps containing valid anchors. No separate per-agent death mask is applied in this objective. The optional auxiliary future-role prediction loss has zero weight in this configuration.

Slot memory and training configuration. During contrastive training, retrieval is performed after refining the first $T_c = 3$ steps of each sampled sequence. The retrieval features are the normalized 128-dimensional trajectory embeddings at that step. Their normalized sum forms the pooled agent key used for coarse screening. At most 50 entries are screened, and the five candidates with the highest mean Hungarian-matched cosine similarities are retained. The selected entry maximizes the stored terminal win indicator $\rho \in \{0, 1\}$; its slots are reused according to the main-text probability with $\tau_m = 10$. Retrieved slots affect subsequent refinement steps, since the current step's assignments are recorded before retrieval. The pre-retrieval local slots are then stored as detached tensors in a FIFO bank of 2,048 entries for sequences with a valid context. Both wins and losses are stored, and repeated replay samples are not deduplicated. The implementation also stores a normalized mean global-state context for fallback retrieval; the Hungarian path uses the pooled agent key instead. Slot memory is used during contrastive learning, whereas decentralized action selection uses the trajectory encoder, role encoder, and individual Q-network without memory retrieval.

Training uses Adam and episode batches. Before TD training, the runner performs 120 trajectory-encoder pretraining updates, predicting the next observation and agent identity, followed by 100 contrastive pretraining updates. Table 3 lists the training and memory settings. The target Q-network, target mixing network, and momentum role encoder are updated after each TD update with coefficient 0.005. The momentum role encoder receives an additional update with the same coefficient after each contrastive update. The TD optimizers decay their learning rates by a factor of 0.98 every 500 training updates.

Table 3: Training and slot-memory hyperparameters of SPAR.

Parameter	Value	Parameter	Value
Optimizer	Adam	Slot count K	3
Replay capacity (episodes)	5,000	Slot refinement iterations	1
Batch size (episodes)	32	Slot temperature τ_s	1.0
TD learning rate	6×10^{-4}	Contrastive loss weight	1.0
Contrastive learning rate	8×10^{-4}	Negative weakening α	1.0
Encoder pretraining learning rate	10^{-3}	Auxiliary gate	Soft weight
Discount factor γ	0.99	Bidirectional gate aggregation	Maximum
Contrastive interval T_d (updates)	200	Future-role loss weight	0.0
Target soft-update coefficient	0.005	Memory enabled	True
Role momentum coefficient	0.005	Memory context length T_c	3
Learning-rate decay interval (updates)	500	Memory capacity (entries)	2,048
Learning-rate decay factor	0.98	Coarse candidate limit	50
Initial exploration ϵ	1.0	Retained candidates M	5
Final exploration ϵ	0.02	Reuse coefficient τ_m	10.0
Exploration annealing (env. steps)	80,000	Candidate refinement	Hungarian
Evaluation interval (env. steps)	10,000	Candidate selection	Win indicator
Episodes per evaluation	32	Retrieval features	Trajectory embeddings
Trajectory pretraining updates	120	Memory replacement	FIFO
Contrastive pretraining updates	100	Role encoder layer normalization	False

Algorithm 1: SPAR based on QMIX

Input: ϕ: agent's trajectory encoder
$\theta, \bar{\theta}$: role encoder and momentum role encoder
K: number of slots
T_d: training interval for updating the contrastive loss
T_c: context length for slot-memory retrieval
n: number of agents
$\mathcal{R}$: trajectory replay buffer
$\mathcal{B}$: slot memory bank
T: maximum time horizon of a learning episode

Initialize all network parameters and initial slot prototypes C_0
Initialize $\mathcal{R}$, $\mathcal{B}$, and training counter $v \leftarrow 0$
Collect warm-up trajectories and pretrain the trajectory and role encoders
for $episode = 1, 2, \ldots$ **do**
  Reset the environment, recurrent hidden states, and previous actions
  for $t = 1, 2, \ldots, T$ *until termination* **do**
    Obtain partial observations $\{o_i^t\}_{i=1}^n$, global state s^t, and available actions
    for $agent\ i = 1, 2, \ldots, n$ **do**
      Calculate the trajectory embedding $e_i^t = f_\phi(o_i^t, a_i^{t-1}, e_i^{t-1})$
      Calculate the role representation $z_i^t = f_\theta(e_i^t)$
      Select a_i^t by masked ϵ-greedy action selection using the role-conditioned Q_i
    end
    Execute joint action $\boldsymbol{a}^t$ and obtain the global reward r^t
  end
  Store the trajectory and its terminal win indicator ρ in $\mathcal{R}$
  Sample a batch of trajectories from $\mathcal{R}$; set $v \leftarrow v + 1$
  if $v \bmod T_d = 0$ **then**
    Recompute trajectory embeddings e_i^t, queries $z_i^t = f_\theta(e_i^t)$, and momentum keys $\bar{z}_i^t = f_{\bar{\theta}}(e_i^t)$
    Initialize pending entries $\mathcal{U} \leftarrow [\,]$
    for *each sampled trajectory* **do**
      Initialize episode-local slots $C \leftarrow C_0$
      for *each sequence step* t **do**
        Refine C from projected embeddings by assignment-weighted aggregation and a slot GRU
        Record assignments P_i^t and their first- and second-ranked slots m_i^t, u_i^t
        if $t = T_c$ **then**
          Save local slots $C^{\mathrm{loc}} \leftarrow C$ and construct context features (q, F)
          Coarse-filter $\mathcal{B}$ by pooled agent features and retain the top-M candidates ranked by Hungarian matching scores
          if *the candidate set* $\mathcal{C}(q, F)$ *is nonempty* **then**
            Select $r^* \leftarrow \arg\max_{r \in \mathcal{C}(q,F)} \rho^r$ and compute p_{reuse}
            Draw $b \sim \mathrm{Bernoulli}(p_{\mathrm{reuse}})$; set $C \leftarrow C^{r^*}$ if $b = 1$
          end
          Append $(q, F, \rho, C^{\mathrm{loc}})$ with valid context to $\mathcal{U}$
        end
      end
    end
    Insert all entries in $\mathcal{U}$ into the FIFO memory $\mathcal{B}$ in batch order
    Construct positive and negative key sets by agreement and disagreement of primary roles
    Compute auxiliary-aware negative weights ω_{ij}^t and optimize the contrastive objective $\mathcal{L}_{\mathrm{ma}}$
    Update the momentum role encoder with coefficient τ_{role}
  end
  Recompute role representations and individual Q-values for the sampled trajectories
  Update the Q-network, mixing network, trajectory encoder, and role encoder using masked TD loss
  Update target Q/mixing networks on schedule and the momentum role encoder with coefficient τ
end