Learning Bilateral Team Formation in Cooperative Multi-Agent Reinforcement Learning

Koorosh Moslemi, Chi-Guhn Lee

Keywords: Team Formation, Stable Matching, Cooperative MARL

Summary

Team formation and the dynamics of team-based learning have drawn significant interest in the context of Multi-Agent Reinforcement Learning (MARL). However, existing studies primarily focus on unilateral groupings, predefined teams, or fixed-population settings, leaving the effects of algorithmic bilateral grouping choices in dynamic populations underexplored. To address this gap, we introduce a framework for learning two-sided team formation in dynamic multi-agent systems. Through this study, we gain insight into what algorithmic properties in bilateral team formation influence policy performance and generalization. We validate our approach using widely adopted multi-agent scenarios, demonstrating competitive performance and improved generalization in most scenarios.

Contribution(s)

1. We study the team formation problem in MARL as a bilateral matching between two disjoint sets of agents. We empirically show a stable matching algorithm results in policies that generalize better to unseen agent compositions compared to those derived from an unstable matching.

   Context: The matching market problem involves matching two disjoint sets based on their preferences. A key property in this domain is *stability* (Gale & Shapley, 1962), which ensures that no agent has an incentive to deviate by seeking alternative matches. Within this domain, the problem of learning preferences has primarily been explored in a bandit framework. While previous research has examined aspects of stable matchings such as centralized versus decentralized (Liu et al., 2020), one-sided versus two-sided preference learning (Zhang & Fang, 2024), and regret analysis in bandit settings, our work shifts the focus to MARL. Specifically, we investigate the performance of stable and unstable matchings in MARL, providing new insights into their comparative effectiveness.

2. We propose an attention-based value decomposition method for team formation, introducing targeted modifications to standard value and mixing networks. This framework enables learning bilateral team formation for a dynamic population of agents.

   Context: The core contribution of our work lies in interpreting attention scores among agents as preferences and leveraging matching algorithms to process these learned preferences. While prior studies, such as GoMARL (Zang et al., 2024) and VAST (Phan et al., 2021), investigate the influence of subgroups on value decomposition within a fixed agent population, our framework relaxes this assumption by only constraining the maximum number of agents. Alternative approaches, such as COPA (Liu et al., 2021), require pre-defined teams during training to bypass learning the team formation process. Others adopt unilateral teaming methods, such as SOG (Shao et al., 2022), which relies on random selection for one side of the matching process.

Learning Bilateral Team Formation in Cooperative Multi-Agent Reinforcement Learning

Koorosh Moslemi[1,†], Chi-Guhn Lee[1]

`koorosh.moslemi@mail.utoronto.ca, chiguhn.lee@utoronto.ca`

[1]**University of Toronto, Canada**

[†] Corresponding author

Abstract

Team formation and the dynamics of team-based learning have drawn significant interest in the context of Multi-Agent Reinforcement Learning (MARL). However, existing studies primarily focus on unilateral groupings, predefined teams, or fixed-population settings, leaving the effects of algorithmic bilateral grouping choices in dynamic populations underexplored. To address this gap, we introduce a framework for learning two-sided team formation in dynamic multi-agent systems. Through this study, we gain insight into what algorithmic properties in bilateral team formation influence policy performance and generalization. We validate our approach using widely adopted multi-agent scenarios, demonstrating competitive performance and improved generalization in most scenarios.

1   Introduction

In recent years, Multi-Agent Reinforcement Learning (MARL) has emerged as a powerful framework for tackling complex decision-making problems that require coordinated actions across multiple autonomous agents. Such systems find application in various domains, from traffic control (Zhang et al., 2024) and autonomous vehicles (Lee et al., 2025) to robotics (Krnjaic et al., 2024). Within this framework, team formation has gained particular attention, as the ability to form, adapt, and coordinate effective teams of agents creates additional possibilities for incorporating valuable group-level learning cues.

In the realm of team formation, value decomposition methods have gained traction in recent years. These methods typically break down the centralized action-value function into individual value functions, each conditioned on the state and actions of a single agent. However, recent works such as VAST (Phan et al., 2021) and GoMARL (Zang et al., 2024) have highlighted performance bottlenecks in such flat decompositions. To address this, they propose a group-wise factorization approach. Despite these advancements, there is no established consensus on how to form these groups. Existing methods have explored various strategies, including leveraging clustering (Phan et al., 2021), heuristic-based approaches (e.g., initially placing all agents in a single group and gradually reallocating them based on predefined rules) (Zang et al., 2024), one-sided matching between two disjoint agent subsets (Shao et al., 2022), and implicit grouping by learning to allocate tasks to agents (Wang et al., 2020) in hierarchical reinforcement learning literature.

In a different research domain, the matching market problem—which involves matching two disjoint sets based on their preferences—has been extensively studied in applications such as college admissions (Gale & Shapley, 1962), labor markets (Roth & Peranson, 1999), and medical residency programs (Roth, 1984). A key property in this domain is stability, which ensures that no agent has an incentive to deviate by seeking alternative matches. Since agents' preferences may be uncer-

tain or unknown depending on the application, significant research has focused on learning these preferences, often within a bandit framework. However, to the best of our knowledge, the role of stability of matching and two-sided preference learning has not been examined in the context of team formation in MARL.

In MARL, nonstationarity poses a fundamental challenge—agents continuously improve their skills over time while the environment itself changes. Learning preferences in such a dynamic setting is significantly more complex than in traditional multi-armed bandit problems, where the reward distribution of each arm remains stationary. A key distinction in MARL is the interdependence between matching and preference learning: the way agents are matched influences the environment dynamics, which in turn affects preference learning, while preferences themselves shape the outcomes of matching. This interplay raises a crucial question: what properties of a matching mechanism facilitate effective learning? To illustrate, consider a heterogeneous team of ground and aerial robots deployed in a search-and-rescue scenario. Ground robots must learn to navigate rough terrain, remove obstacles, and transport supplies, while aerial robots focus on mapping, scouting, and relaying communication. For effective operation, these robots must form cooperative ground-aerial pairs while simultaneously learning their preferences—such as camera resolution, and low-latency communication. Since both ground and aerial robots are continuously learning, they may develop conflicting perspectives on optimal pairings, despite being trained for a common objective. A matching mechanism aggregates differing views, influencing the overall learning outcome. This leads to an important inquiry: does a property like stability in matching (i.e., reducing the likelihood of agents frequently switching partners) improve resulting policies? To address this question, we make the following contributions:

- We study the team formation problem in MARL as a bilateral matching between two disjoint sets of agents. We empirically show a stable matching algorithm results in policies that generalize better to unseen agent compositions compared to those derived from an unstable matching.

- We propose an attention-based value decomposition method for team formation, introducing targeted modifications to standard value and mixing networks. This framework enables learning bilateral team formation for a dynamic population of agents.

2   Related Work

Team Formation   Zang et al. (2024) introduced GoMARL, a grouping method designed for a fixed population of agents. In this method, all agents are initially in one group, and they're gradually reassigned to other groups based on a heuristic that evaluates their fitness to the current group. By learning an adaptive group structure, GoMARL learns the state-action value function for groups of agents, addressing performance bottlenecks in flat decomposition methods that rely on individual value functions. Similarly, our method employs group-wise value decomposition. However, unlike GoMARL, the algorithms we explore are not tied to a specific learning paradigm (i.e., they can be applied to both policy gradient and value-based methods). Additionally, our approach accommodates a variable number of agents at execution, provided that the maximum number is predetermined during training. Liu et al. (2021) proposed COPA, which introduced a coach agent with global information to periodically broadcast strategies to a dynamic number of agents. COPA departs from the common decentralized execution (DE) assumption, arguing that it is too restrictive for complex tasks. Our work similarly challenges this assumption by enforcing structured group formation during execution. However, unlike COPA which relies on a predefined set of teams during training, we learn inter-agent preferences that lead to matching agents and forming teams. SOG (Shao et al., 2022) explored the idea of matching two disjoint sets of agents—conductors and non-conductors. Conductor agents, either randomly selected or elected, send group invitations to non-conductor agents, who then choose among multiple invitations at random. In contrast, our work focuses on bilateral team formation, where matching is driven by mutual preferences rather than unilateral selection. Within the DE paradigm, CollaQ (Zhang et al., 2020) and MIPI (Ye & Lu, 2024) addressed generalization challenges in dynamic team formation. Similarly, our work aims to develop a method that results

in moderately close performance between training and execution as the number of agents increases at execution. However, we specifically examine how the properties of matching algorithms, such as stability versus instability, affect team formation in a centralized execution setting.

Matching and Learning Preferences   The problem of learning preferences in matching markets is often studied within a multi-armed bandit framework, where agents have uncertain (Aziz et al., 2020) or unknown preferences over arms, while the arms' preferences over agents are sometimes assumed to be known (Liu et al., 2020; Hosseini et al., 2024). The primary objective in this literature is to design algorithms that minimize regret for each agent. Liu et al. (2020) formalize both centralized and decentralized bandit learning for matching markets. More recently, Zhang & Fang (2024) extend this framework to settings where preferences on both sides are unknown. Wang et al. (2022) and Kong & Li (2024) further generalize the problem from one-to-one to many-to-one matching. Similarly, we investigate a many-to-one matching problem where preferences on both sides are unknown. However, in our case, matching occurs among agents collaboratively optimizing the expected discounted sum of rewards in a MARL setting.

3   Background

3.1   Entity-wise Dec-POMDP

We study the *decentralized partially observable Markov decision process* (Dec-POMDP) (Oliehoek et al., 2016) with entities (Schroeder de Witt et al., 2019) described as $(\mathbf{S}, \mathbf{U}, \mathbf{O}, P, r, \mathcal{E}, \mathcal{A}, \mu)$. $\mathcal{E}$ represents entities in the environment, either agents or non-agents (i.e., $\mathcal{A} \subseteq \mathcal{E}$). Unlike the set of agents $\mathcal{A}$, non-agent entities cannot be controlled by learning policies (e.g., obstacles, enemies with fixed behavior). $\mathbf{s} \in \mathbf{S}$ denotes the global state containing state representation for entities s^e (i.e., $\mathbf{s} = \{s^e \mid e \in \mathcal{E}\} \in \mathbf{S}$). Each agent's partial observability $o^a \in \mathbf{O}$ is defined as $\{s^e \mid \mu(s^a, s^e) = 1, e \in \mathcal{E}, a \in \mathcal{A}\}$ where s^a is the state representation for agent $a \in \mathcal{A}$, and $\mu(s^a, s^e) \in \{0, 1\}$ is the binary observability mask. In other words, $\mu(s^a, s^e) = 1$ means agent $a \in \mathcal{A}$ observes entity $e \in \mathcal{E}$. $\mathbf{U}$ is the set of joint actions of all agents. Additionally, P and r denote the state transition and reward functions, respectively.

To deal with a dynamic population of agents, a multi-head attention module, denoted as $\mathrm{MHA}(\mathcal{A}, eFF(\mathbf{X}^{\mathcal{E}}), \mathbf{M})$, integrates information not blocked by masks. $\mathbf{X}^{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}| \times d}$ is the entity representation matrix where d is the dimensionality of the input. $eFF(.)$ denotes an entity-wise feedforward layer, which is a standard fully connected layer that applies identical transformations to all input entities. $\mathbf{M} \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{E}|}$ is the mask matrix which specifies which entities can be queried by agents. For more details on the attention module please refer to (Iqbal et al., 2021b). Additionally, we use the subscript t to denote the timestep and define the discount factor as $\gamma \in [0, 1)$.

3.2   Value Decomposition

Value decomposition methods (Sunehag et al., 2017) decompose the centralized action-value function $Q^{\mathrm{tot}}(\mathbf{s}, \mathbf{u})$ into simpler value functions (e.g., individual value functions that are conditioned on states and actions of one agent). This decomposition addresses the exponential growth of the joint action space with the number of agents and enables decentralized action selection. The centralized action-value function is defined as:

$$Q^{\mathrm{tot}}(\mathbf{s}, \mathbf{u}) := r(\mathbf{s}, \mathbf{u}) + \gamma \mathbb{E}\left[\max Q^{\mathrm{tot}}(\mathbf{s}', \cdot) \mid s' \sim P(\cdot \mid \mathbf{s}, \mathbf{u})\right].$$

QMIX (Rashid et al., 2018) uses a monotonic mixing function f_{mix} that combines individual utilities to approximate Q^{tot}. The monotonicity of f_{mix} ensures an increase in the value of any agent Q^a for its action u^a leads to an increase in Q^{tot} for joint actions $\mathbf{u} \in \mathbf{U}$. Consequently, composing greedy individual actions with respect to individual value functions results in a greedy joint action with

respect to Q^{tot}. In other words, f_{mix} is used as follows:

$$Q_\theta^{\text{tot}}\left(\boldsymbol{\tau}_t, \mathbf{u}_t\right) \approx f_{mix}\left(Q^1\left(\tau_t^1, u_t^1; \theta_Q\right), \ldots, Q^{|\mathcal{A}|}\left(\tau_t^{|\mathcal{A}|}, u_t^{|\mathcal{A}|}; \theta_Q\right); \theta\right),$$
$$\theta = h(\boldsymbol{s}_t; \theta_h).$$

where $Q_\theta^{\text{tot}}\left(\boldsymbol{\tau}_t, \mathbf{u}_t\right) \approx Q^{\text{tot}}\left(\mathbf{s}_t, \mathbf{u}_t\right)$ is the parametrized centralized value function. This function uses trajectories $\boldsymbol{\tau}_t := \{\tau_t^a\}_{a \in \mathcal{A}}$, where $\tau_t^a := (o_0^a, u_0^a, \ldots, o_t^a)$ denotes history of an agent's actions and observations, as a proxy for global state. h is a hyper-network (Ha et al., 2016) conditioned on the global state and parametrized by θ_h. In practice, the monotonicity assumption is ensured by constraining θ to be positive using techniques such as applying the absolute value function or softmax normalization. To learn the Q-function, the following objective is minimized:

$$\mathcal{L}_Q(\theta) := \mathbb{E}\left[\left(y_t^{\text{tot}} - Q_\theta^{\text{tot}}\left(\boldsymbol{\tau}_t, \mathbf{u}_t\right)\right)^2 \mid (\boldsymbol{\tau}_t, \mathbf{u}_t, r_t, \boldsymbol{\tau}_{t+1}) \sim \mathcal{D}\right], \tag{1}$$
$$y_t^{\text{tot}} := r_t + \gamma Q_{\bar{\theta}}^{\text{tot}}\left(\boldsymbol{\tau}_{t+1}, \arg\max Q_\theta^{\text{tot}}\left(\boldsymbol{\tau}_{t+1}, \cdot\right)\right).$$

where $Q_{\bar{\theta}}^{\text{tot}}$ is a target network (Mnih et al., 2015) with weights $\bar{\theta}$ that are copied from θ periodically to stabilize the regression target y_t^{tot}. $\mathcal{D}$ is a replay buffer (Lin, 1992) containing experiences of all agents collected by a policy for exploration.

4 Method

In this section, we detail the modifications made to agents' utility (a.k.a., value) and mixing networks within a value decomposition framework. We build on REFIL (Iqbal et al., 2021a), as it already incorporates an attention mechanism capable of handling a dynamic number of agents. In short, these modifications include changes to the architecture of the models (e.g., adding encoder-decoder networks to enforce group structures) and training objectives (e.g. encouraging similarity within groups and diversity among groups).

4.1 Agent Utility Network

As illustrated in Figure 1a, this network integrates information across entities with an attention module $\text{MHA}(\mathcal{A}, eFF(\tilde{\mathbf{X}}^{\mathcal{E}}), (\mathbf{M}^\mu, \mathbf{M}_O^\mu, \mathbf{M}_I^\mu))$[1]. $\mathbf{M}_O^\mu$ and $\mathbf{M}_I^\mu$ are random binary attention masks of size $|\mathcal{A}| \times |\mathcal{E}|$ restricted by the partial observability mask $\mathbf{M}^\mu$. These masks enable counterfactual reasoning by introducing randomly created complementary subsets of entities denoted by I and O. $\tilde{\mathbf{X}}^{\mathcal{E}}$ of size $|\mathcal{E}| \times (d+3)$ is our augmented version of $\mathbf{X}^{\mathcal{E}}$ with a one-hot vector specifying the type of the entity (i.e., leader, follower, or non-agent entity). We augment the entity representation matrix $\mathbf{X}^{\mathcal{E}}$ by partitioning the set of agents $\mathcal{A}$ into the set of leader agents (i.e., denoted by $\mathcal{L}$) and follower agents (i.e., denoted by $\mathcal{F}$) as follows:

$$\mathcal{A} = \mathcal{L} \cup \mathcal{F}, \ \mathcal{L} \cap \mathcal{F} = \emptyset, \ |\mathcal{L}| \leq |\mathcal{F}|.$$

Our goal is to learn matching between agents in $\mathcal{F}$ and agents in $\mathcal{L}$. To this end, we use attention weights returned by $\text{MHA}(\mathcal{A}, eFF(\tilde{\mathbf{X}}^{\mathcal{E}}), \mathbf{1}^{|\mathcal{A}| \times |\mathcal{E}|})$ as preferences among agents for team formation.

Different choices for preference-based matching are discussed in Section 4.3. From now on, we assume the groups are denoted with set $\mathcal{G}$.

Inspired by the utility network architecture proposed by Zang et al. (2024), we add an encoder-decoder module to the agent utility network. As illustrated in Figure 1a, the group-aware encoder $f_e(.; \theta_e)$ embeds agents' hidden states such that embeddings corresponding to agents within a group are similar while embeddings corresponding to agents from distinct groups are different. Details for training the encoder $f_e(.; \theta_e)$ are in Section 4.4. The decoder is a linear transformation that

---

[1]Technically, $\tilde{\mathbf{X}}^{\mathcal{E}}$ is repeated to adjust for the three masks. However, we abuse the notation in writing.
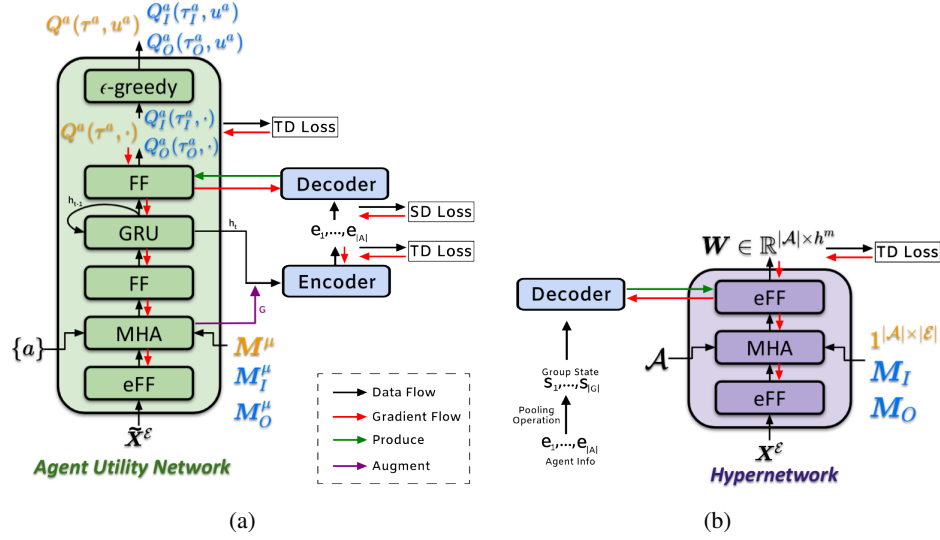
Figure 1: Modifications to agent utility network and hypernetwork. (a) An encoder-decoder structure is integrated to incorporate group-related information into the action selection process. The type of utility used in Equations 2 and 3 (i.e., Q^a, Q_I^a, Q_O^a) depends on the input mask of the multi-head attention (MHA) module as color-coded in Figure 1a (b) A group-aware hypernetwork is employed to generate the weights for the mixing network f_{mix}. The decoder used here is different from Figure 1a as they generate weights for different layers. However, the input embeddings $e_1, \cdots, e_{|\mathcal{A}|}$ passed to both decoders are the same.

generates parameters of the last feedforward layer in the agent's utility network. **These changes have three key benefits: 1) enabling learning of bilateral team formation for a dynamic population of agents; 2) comparing the effect of algorithmic choices for matching on performance and generalization; 3) controlling the composition of new agents (i.e., adding new agents as followers or leaders) in unseen scenarios.**

4.2 Hypernetwork

Similar to the agent's utility network, $\mathbf{X}^{\mathcal{E}}$ is fed into an entity-wise feedforward layer followed by an attention module $\text{MHA}(\mathcal{A}, eFF(\mathbf{X}^{\mathcal{E}}), (\mathbf{1}^{|\mathcal{A}| \times |\mathcal{E}|}, \mathbf{M}_O, \mathbf{M}_I))$. Note that all masks ignore partial observability (i.e., replacing $\mathbf{M}^\mu$ with an all-ones matrix $\mathbf{1}^{|\mathcal{A}| \times |\mathcal{E}|}$ and not restricting $\mathbf{M}_O^\mu, \mathbf{M}_I^\mu$ with $\mathbf{M}^\mu$) as the hypernetwork is used during centralized training. As shown in Figure 1b, embeddings of agents' hidden states generated by the encoder $f_e(.; \theta_e)$ in the agent's utility network are passed through a group-aware pooling operation that generates group-wise states. These states are then fed into a decoder that generates the parameters for the last feedforward layer in the hypernetwork. **This modification enables factorization of joint action-values by considering group information.**

4.3 Matching Algorithms

The algorithms we consider in this section require: 1) An inter-agent preference matrix of size $|\mathcal{A}| \times |\mathcal{A}|$ where each row quantifies an agent's preferences on other agents; 2) a prior on the number of leader agents and the size of each team. The inter-agent preference matrix is learned as part of our learning framework by the attention module. Moreover, we explain our choices for the priors in Section 5. The algorithms are described as follows:

Order Oriented Matching (OOM) Algorithm 1 only considers the order of the preferences using a *deferred acceptance* (DA) mechanism and is also known as the many-to-one variant of the stable matching algorithm (Gale & Shapley, 1962). The DA mechanism ensures that the matching out-

come is stable, meaning no two agents would prefer to be paired with each other over their current assignments. The algorithm begins with leaders proposing to their top-choice followers until they reach their team size limit. Meanwhile, each follower tentatively accepts the best-received proposal, rejecting any inferior ones. Rejected leaders then propose their next choice. This process repeats until all the leaders have their teams.

Score Oriented Matching (SOM)   In this algorithm, scores computed by the attention module influence the matching outcome. The core idea is that followers iterate through leaders to find the best free leader based on the mutual score, as shown in Algorithm 2. Unlike OOM, this algorithm does not employ the DA mechanism, making it unstable.

4.4   Training

In addition to $\mathcal{L}_Q$ in Equation 1, an auxiliary loss $\mathcal{L}_{aux}$ is used to factorize Q-function with $2|\mathcal{A}|$ factors (i.e., Q_I^a and Q_O^a illustrated in Figure 1a) by replacing Q^{tot} with Q_{aux}^{tot}:

$$Q^{\text{tot}} = f_{mix}\left(Q^1, \ldots, Q^{|\mathcal{A}|}; h\left(\mathbf{s}; \theta_h, \mathbf{1}^{|\mathcal{A}| \times |\mathcal{E}|}, \boldsymbol{E}\right)\right), \tag{2}$$

$$\approx Q_{\text{aux}}^{\text{tot}} = f_{mix}\left(Q_I^1, \ldots, Q_I^{|\mathcal{A}|}, Q_O^1, \ldots, Q_I^{|\mathcal{A}|} ; h\left(\mathbf{s}; \theta_h, \boldsymbol{M}_I, \boldsymbol{E}\right), h\left(\mathbf{s}; \theta_h, \boldsymbol{M}_O, \boldsymbol{E}\right)\right). \tag{3}$$

In the above equations, h denotes the hypernetwork explained in Section 4.2 and $\boldsymbol{E} = \{e_1, \cdots, e_{|\mathcal{A}|}\}$ is the set of embeddings generated by the encoder $f_e(.; \theta_e)$ introduced in Section 4.1. We train the embeddings using the following loss function similar to (Zang et al., 2024):

$$\mathcal{L}_{SD}\left(\theta_e\right) = \mathbb{E}_{\mathcal{B}}\left(\sum_{i \neq j} I(i, j) \cdot \text{cosine}\left(f_e\left(h^i; \theta_e\right), f_e\left(h^j; \theta_e\right)\right)\right), \tag{4}$$

$$\text{where } I(i, j) = \begin{cases} -1, & a_i, a_j \in g_k, g_k \in \mathcal{G}. \\ 1, & a_i \in g_k, a_j \in g_l, k \neq l, g_k, g_l \in \mathcal{G}. \end{cases}$$

In the above equation, h^i and h^j denote hidden states of agents a_i and a_j respectively. This loss ensures the hidden state embeddings of agents within the same group g_k to be similar. At the same time, it prevents all agents from being alike by encouraging diversity between agents $a_i \in g_k, a_j \in g_l$ from different groups g_k, g_l. The final loss is defined with a tradeoff constant $\lambda \in [0, 1]$ as follows:

$$\mathcal{L}_{TD} = (1 - \lambda)\mathcal{L}_Q + \lambda\mathcal{L}_{aux},$$
$$\mathcal{L} = \mathcal{L}_{TD} + \mathcal{L}_{SD}.$$

5   Experimental Setup

Problem Setting Instantiation   We evaluate our approach in the StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) to keep our experiments comparable with the previous works. We train $3-5$ agents with different matching algorithms in the customized scenarios introduced by Ye & Lu (2024). The scenario names correspond to specific unit types; for example, SZ represents Stalkers and Zealots, CSZ includes Colossi, Stalkers, and Zealots, and MMM refers to Marines, Marauders, and Medivacs. In evaluation, we use $6 - 8$ agents with varying leader-follower composition. We report the mean and standard deviation of the winning rate for five random seeds.

Implementation Details   We do not tune the default hyperparameters in REFIL to ensure a fair comparison with baseline approaches (e.g., $\lambda = 0.5$).

In practice, we use the modified hypernetwork only to generate the weight of the first layer of the mixing networks in Equations 2 and 3. Additionally, we use max pooling to generate group states

in Figure 1b. We use four heads for the attention module in the agent's utility network and unify the attention scores with max pooling across different heads. For augmenting hidden states in Figure 1a, we concatenate a one-hot vector of random mapping of group number to a set of size four (i.e., the maximum number of supported groups in our experiment). During training, we set $|\mathcal{L}| = 2$ and enforce a team size balancing strategy across all matching algorithms, ensuring equal or nearly equal team sizes among leaders. This strategy remains valid during evaluation, where the number of leaders $|\mathcal{L}|$ varies within the range $[2, 4]$. Furthermore, in both training and evaluation, the first $|\mathcal{L}|$ agents are designated as leaders.

6 Results and Discussion

Table 1: Winning rate on SMAC after $2M$ timesteps

Tasks	Algorithms	Training	Evaluation		
		$3 - 5$	6	7	8
SZ	MIPI	0.659 ± 0.02	0.453 ± 0.08	0.404 ± 0.062	0.276 ± 0.076
	REFIL	$\mathbf{0.674} \pm 0.038$	0.441 ± 0.103	0.352 ± 0.078	0.236 ± 0.103
	AQMIX	0.528 ± 0.044	0.343 ± 0.105	0.291 ± 0.084	0.182 ± 0.058
	CollaQ	0.588 ± 0.03	0.366 ± 0.086	0.314 ± 0.076	0.198 ± 0.097
	MAPPO	0.256 ± 0.01	0.129 ± 0.019	0.148 ± 0.031	0.036 ± 0.015
	OOM*(ours)	0.624 ± 0.010	$\mathbf{0.531} \pm 0.037$	$\mathbf{0.442} \pm 0.044$	$\mathbf{0.364} \pm 0.019$
	SOM*(ours)	0.639 ± 0.007	0.475 ± 0.049	0.412 ± 0.041	0.308 ± 0.022
CSZ	MIPI	0.548 ± 0.032	0.42 ± 0.102	0.297 ± 0.112	0.261 ± 0.09
	REFIL	$\mathbf{0.568} \pm 0.027$	0.348 ± 0.057	0.229 ± 0.053	0.164 ± 0.06
	AQMIX	0.509 ± 0.054	0.323 ± 0.096	0.216 ± 0.101	0.152 ± 0.071
	CollaQ	0.459 ± 0.061	0.362 ± 0.13	0.267 ± 0.099	0.231 ± 0.095
	MAPPO	0.248 ± 0.037	0.12 ± 0.029	0.06 ± 0.028	0.054 ± 0.013
	OOM*(ours)	0.534 ± 0.007	0.399 ± 0.017	$\mathbf{0.317} \pm 0.005$	$\mathbf{0.329} \pm 0.025$
	SOM*(ours)	0.539 ± 0.010	$\mathbf{0.486} \pm 0.092$	0.283 ± 0.007	0.181 ± 0.012
MMM	MIPI	0.548 ± 0.023	0.495 ± 0.054	$\mathbf{0.447} \pm 0.041$	$\mathbf{0.467} \pm 0.067$
	REFIL	$\mathbf{0.605} \pm 0.057$	0.437 ± 0.118	0.329 ± 0.171	0.224 ± 0.163
	AQMIX	0.501 ± 0.036	0.447 ± 0.043	0.344 ± 0.071	0.251 ± 0.089
	CollaQ	0.589 ± 0.027	$\mathbf{0.513} \pm 0.07$	0.423 ± 0.026	0.286 ± 0.083
	MAPPO	0.289 ± 0.097	0.32 ± 0.102	0.25 ± 0.063	0.275 ± 0.098
	OOM*(ours)	0.586 ± 0.016	0.443 ± 0.045	0.394 ± 0.021	0.332 ± 0.093
	SOM*(ours)	0.498 ± 0.009	0.318 ± 0.017	0.219 ± 0.048	0.165 ± 0.035

Table 1 presents the performance of the best compositions of SOM and OOM compared to prior methods. The results for MIPI, REFIL, CollaQ, AQMIX (Iqbal et al., 2020), and MAPPO (Yu et al., 2022) are adopted from a similar experimental setup by Ye & Lu (2024). Detailed performance breakdowns for each composition are provided in Tables 2, 3, 4. In this work, the added components are specifically designed to operate in tandem with the matching algorithms, forming a tightly coupled framework. As such, ablation of individual modules (e.g., retaining the encoder-decoder structure while removing matching) is not informative. We instead use REFIL as a meaningful ablation-style baseline, which lacks both the matching logic and architectural components, allowing us to evaluate the effect of their integration. Our findings indicate that in **6 out of 9** evaluation

scenarios, the best compositions in our methods outperform the baselines, while their performance remains comparable in training scenarios. Notably, in **26 out of 27** evaluation compositions OOM consistently outperforms SOM. We hypothesize that the superior generalization capability of OOM stems from its stability property. Specifically, by ensuring that no agent has an incentive to deviate from the final matching, OOM may discourage inefficient policy switching caused by frequent group changes.

A promising direction for future work is to investigate the performance of SOM and OOM in the context of *distracted attention issue*. This phenomenon suggests that increasing agents' sight range leads attention mechanisms such as REFIL's to focus on irrelevant context, thereby degrading team performance (Shao et al., 2023). Thus, future work can study whether OOM—which uses only the relative order of scores—is more robust to such distractions compared to SOM, which directly relies on attention scores. Future research can also further investigate the comparative performance of OOM and SOM by incorporating a more robust attention mechanism, such as differential attention (Ye et al., 2024). Additionally, developing a principled approach for tuning $|\mathcal{L}|$ and selecting leaders, inspired by methods proposed in (Shao et al., 2022), remains an open direction for future work.

7 Conclusion

In this work, we introduced a framework to study algorithmic choices for the dynamic grouping of agents. In particular, we investigated two bilateral matching methods and empirically concluded that the stability of the matching algorithm results in teams that show better generalization in unseen tasks.

Acknowledgments

A Algorithms

Algorithm 1 Order Oriented Matching

1: Initialize all leaders and followers to be free.
2: Determine the number of spots on each leader's team.
3: **while** there is a leader l who has not proposed to every follower and has an empty spot in its team **do**
4:     Choose such a leader l
5:     Let f be the highest-ranked follower in l's preference list to whom l has not yet proposed
6:     **if** f is on no team **then**
7:         Add f to the team of l
8:     **else**
9:         Suppose f be on team of l'
10:         **if** f prefers l' to l **then**
11:             No follower will be recruited in l's team
12:         **else**
13:             f prefers l to l'
14:             f will be added to the team of l, l' will have one more empty spot in its team
15:         **end if**
16:     **end if**
17: **end while**

Algorithm 2 Score Oriented Matching

1: Let S be a score matrix of size $|\mathcal{A}| \times |\mathcal{A}|$ and $c(l)$ denote team capacity for each leader $l \in L$
2: Initialize an empty team list $\mathcal{G}(l) \leftarrow \emptyset$ for each leader $l \in \mathcal{L}$
3: **for** each follower $f \in \mathcal{F}$ **do**
4:     Initialize best_score $\leftarrow -\infty$
5:     Initialize best_leader $\leftarrow$ None
6:     **for** each leader $l \in L$ **do**
7:         **if** $|\mathcal{G}(l)| < c(l)$ **then**   ▷ Check if leader l has capacity
8:             Compute mutual_score $\leftarrow S[f][l] + S[l][f]$
9:             **if** mutual_score $>$ best_score **then**
10:                 best_score $\leftarrow$ mutual_score
11:                 best_leader $\leftarrow l$
12:             **end if**
13:         **end if**
14:     **end for**
15:     **if** best_leader $\neq$ None **then**
16:         Add f to $\mathcal{G}$(best_leader)
17:     **end if**
18: **end for**

B   Extended Results

Table 2: Winning rate of OOM vs SOM under different compositions in SZ scenario

Algorithms	Leaders	Training	Evaluation		
		$3 - 5$	6	7	8
OOM	2	0.624 ± 0.010	$\mathbf{0.531} \pm 0.037$	$\mathbf{0.442} \pm 0.044$	0.329 ± 0.006
	3	-	0.478 ± 0.037	0.433 ± 0.050	0.345 ± 0.004
	4	-	-	-	$\mathbf{0.364} \pm 0.019$
SOM	2	$\mathbf{0.639} \pm 0.007$	0.490 ± 0.038	0.367 ± 0.039	0.300 ± 0.018
	3	-	0.475 ± 0.049	0.412 ± 0.041	0.308 ± 0.022
	4	-	-	-	0.300 ± 0.013

Table 3: Winning rate of OOM vs SOM under different compositions in CSZ scenario

Algorithms	Leaders	Training	Evaluation		
		$3 - 5$	6	7	8
OOM	2	0.534 ± 0.007	0.399 ± 0.017	$\mathbf{0.317} \pm 0.005$	0.297 ± 0.015
	3	-	0.390 ± 0.020	0.315 ± 0.017	$\mathbf{0.329} \pm 0.025$
	4	-	-	-	0.321 ± 0.023
SOM	2	$\mathbf{0.539} \pm 0.010$	0.387 ± 0.073	0.267 ± 0.010	0.173 ± 0.017
	3	-	$\mathbf{0.486} \pm 0.092$	0.283 ± 0.007	0.181 ± 0.012
	4	-	-	-	0.173 ± 0.010

Table 4: Winning rate of OOM vs SOM under different compositions in MMM scenario

Algorithms	Leaders	Training 3 − 5	Evaluation		
			6	7	8
OOM	2	$\mathbf{0.586} \pm 0.016$	$\mathbf{0.443} \pm 0.045$	$\mathbf{0.394} \pm 0.021$	$\mathbf{0.332} \pm 0.093$
	3	-	0.440 ± 0.040	0.360 ± 0.031	0.301 ± 0.091
	4	-	-	-	0.278 ± 0.090
SOM	2	0.498 ± 0.009	0.318 ± 0.017	0.219 ± 0.048	0.165 ± 0.035
	3	-	0.305 ± 0.017	0.199 ± 0.039	0.159 ± 0.028
	4	-	-	-	0.140 ± 0.035

References

Haris Aziz, Péter Biró, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. Stable matching with uncertain linear preferences. *Algorithmica*, 82:1410–1433, 2020.

David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

David Ha, Andrew M Dai, and Quoc V Le. Hypernetworks. In *International Conference on Learning Representations*, 2016.

Hadi Hosseini, Sanjukta Roy, and Duohan Zhang. Putting gale & shapley to work: Guaranteeing stability through learning. *arXiv preprint arXiv:2410.04376*, 2024.

Shariq Iqbal, Christian A Schroeder de Witt, Bei Peng, Wendelin Böhmer, Shimon Whiteson, and Fei Sha. Ai-qmix: Attention and imagination for dynamic multi-agent reinforcement learning. *arXiv preprint arXiv:2006.04222*, 2020.

Shariq Iqbal, Christian A Schroeder De Witt, Bei Peng, Wendelin Böhmer, Shimon Whiteson, and Fei Sha. Randomized entity-wise factorization for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4596–4606. PMLR, 2021a.

Shariq Iqbal, Christian A Schroeder de Witt, Bei Peng, Wendelin Böhmer, Shimon Whiteson, and Fei Sha. Supplementary material: Randomized entity-wise factorization for multi-agent reinforcement learning. *International Conference on Machine Learning*, 2021b.

Fang Kong and Shuai Li. Improved bandits in many-to-one matching markets with incentive compatibility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13256–13264, 2024.

Aleksandar Krnjaic, Raul D Steleac, Jonathan D Thomas, Georgios Papoudakis, Lukas Schäfer, Andrew Wing Keung To, Kuan-Ho Lao, Murat Cubuktepe, Matthew Haley, Peter Börsting, et al. Scalable multi-agent reinforcement learning for warehouse logistics with robotic and human co-workers. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 677–684. IEEE, 2024.

Jonathan W Lee, Han Wang, Kathy Jang, Nathan Lichtlé, Amaury Hayat, Matthew Bunting, Arwa Alanqary, William Barbour, Zhe Fu, Xiaoqian Gong, et al. Traffic control via connected and automated vehicles (cavs): An open-road field experiment with 100 cavs. *IEEE Control Systems*, 45(1):28–60, 2025.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8:293–321, 1992.

Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Anima Anandkumar. Coach-player multi-agent reinforcement learning for dynamic team composition. In *International Conference on Machine Learning*, pp. 6860–6870. PMLR, 2021.

Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, pp. 1618–1628. PMLR, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

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.

T Rashid, M Samvelyan, C Schroeder de Witt, G Farquhar, J Foerster, and S Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *35th International Conference on Machine Learning (ICML 2018)*. Journal of Machine Learning Research, 2018.

Alvin E Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of political Economy*, 92(6):991–1016, 1984.

Alvin E Roth and Elliott Peranson. The redesign of the matching market for american physicians: Some engineering aspects of economic design. *American economic review*, 89(4):748–780, 1999.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Christian Schroeder de Witt, Jakob Foerster, Gregory Farquhar, Philip Torr, Wendelin Boehmer, and Shimon Whiteson. Multi-agent common knowledge reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Jianzhun Shao, Zhiqiang Lou, Hongchang Zhang, Yuhang Jiang, Shuncheng He, and Xiangyang Ji. Self-organized group for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5711–5723, 2022.

Jianzhun Shao, Hongchang Zhang, Yun Qu, Chang Liu, Shuncheng He, Yuhang Jiang, and Xiangyang Ji. Complementary attention for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 30776–30793. PMLR, 2023.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020.

Zilong Wang, Liya Guo, Junming Yin, and Shuai Li. Bandit learning in many-to-one matching markets. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2088–2097, 2022.

Deheng Ye and Zongqing Lu. Mutual-information regularized multi-agent policy iteration. *Advances in Neural Information Processing Systems*, 36, 2024.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024.

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. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Automatic grouping for efficient cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. Multi-agent collaboration via reward attribution decomposition. *arXiv preprint arXiv:2010.08531*, 2020.

YiRui Zhang and Zhixuan Fang. Decentralized two-sided bandit learning in matching market. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

Yuhang Zhang, Zhiyao Zhang, Marcos Quiñones-Grueiro, William Barbour, Clay Weston, Gautam Biswas, and Daniel Work. Field deployment of multi-agent reinforcement learning based variable speed limit controllers. *arXiv preprint arXiv:2407.08021*, 2024.