Adaptive Search in Cooperative Multi-Agent Reinforcement Learning with Parameter Sharing

Yurui Li
College of Coumpter Science and Technology
Zhejiang University
Hangzhou, China
liyr@zju.edu.cn

Li Zhang
College of Coumpter Science and Technology
Zhejiang University
Hangzhou, China
zhangli85@zju.edu.cn

Shijian Li
College of Coumpter Science and Technology
Zhejiang University
Hangzhou, China
shijianli@zju.edu.cn

***Abstract*—Parameter sharing is a widely used approach to tackling the search challenges in the joint observation-action space within Multi-Agent Reinforcement Learning (MARL). Despite its popularity, the effectiveness of parameter sharing varies across different tasks. While previous efforts have sought to enhance the applicability of parameter sharing, none have provided a theoretical analysis of its underlying mechanism. In this study, we demonstrate that the effectiveness of parameter sharing stems from its ability to solidify the search space of MARL methods in the joint observation-action space. However, this solidification isn't applicable universally, rendering it ineffective in certain scenarios. To address this limitation, we propose a novel framework that enables parameter sharing to adaptively determine the search subspace based on specific task. We implement this framework with two representative MARL methods and evaluate their performance across two widely used MARL testbeds. The experimental results confirm that our framework can dynamically adjust the search subspace across tasks, thereby enhancing the performance of the original methods.**

***Index Terms*—MARL, Cooperation**

I. Introduction

Multi-agent reinforcement learning (MARL) has proven effective in various real-world cooperative scenarios, such as autonomous driving [1], traffic management [2], and multiplayer games [3]. However, achieving effective cooperation among agents is difficult due to several significant challenges. The most notable challenges is the exponential growth of the joint action-observation space as the number of agents increases, which places substantial demands on the MARL method's ability to efficiently explore such a vast space. To mitigate this issue, parameter sharing of policy networks has become a widely adopted technique [4]–[8]. Parameter sharing can enhance training efficiency by increasing the quantity of available trajectories for each agent, thus stabilizing the training process [5] and reducing the number of trainable parameters, which aids in achieving convergence. This approach has been particularly successful in cooperative MARL tasks, such as the StarCraft II micromanagement challenge (SMAC) [9] tasks. However, recent studies [4], [10] have highlighted potential drawbacks of parameter sharing in certain contexts. For instance, in the Google Research Football (GRF) tasks [11], parameter sharing may lead to undesirable agent behavior, such as all agents competing for the ball, resulting in similar actions across agents, limiting exploration, and ultimately degrading performance. This phenomenon indicates that the benefits of parameter sharing are not universally applicable across all tasks, and its effectiveness can vary depending on the specific nature of the tasks.

In recent years, substantial efforts have been dedicated to balancing the effectiveness and limitations of parameter sharing, with the primary aim of improving its scope of application. These approaches can be broadly categorized into three groups. The first category includes methods that incorporate discriminative information into agent observations [6]–[8]. In these approaches, discriminative information, often derived from agent IDs, is introduced to differentiate the characteristics of individual agents' observations, thereby enhancing the diversity of agent behavior. The second category focuses on increasing diversity within shared policy networks [4], [10]. These methods typically achieve greater diversity by modifying policy network structures or applying reward shaping techniques, which help prevent homogeneity in agent behavior and foster more effective cooperation. The third category introduces selectively shared policy networks [12]–[16]. These approaches differentiate between sharing and non-sharing policies based on high-level concepts, such as agent roles or subtasks, allowing for more tailored and context-specific cooperation. Despite the progress made in these works, a comprehensive theoretical understanding of the mechanisms underlying the effectiveness or limitations of parameter sharing remains lacking. A deeper theoretical analysis of parameter sharing is essential for enhancing its effectiveness

This work was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No.LD24F030002.

Corresponding Author. Email: shijianli@zju.edu.cn.

across a broader range of tasks.

In this study, we present a comprehensive theoretical analysis, aiming to uncover the mechanisms underlying the effectiveness and limitations of parameter sharing. Our analysis reveals that the effectiveness and limitations of parameter sharing both arise from its solidification for the search space of MARL methods in the joint observation-action space during training, thereby simplifying the exploration process. This solidification occurs through a key phenomenon called **policy homogenization**, where agents tend to select the same action in response to similar observations. As a result, parameter sharing constrains the search space of MARL methods to a subspace characterized by high action similarity across agents during training. To quantify this solidification effect, we introduce a metric called the Rate of Same Action (**RSA**), which divides the joint observation-action space into a complete set of mutually exclusive subspaces from the perspective of the action similarity of multi-agent. The values of RSA can be enumerated, with each value corresponding to a distinct subspace in the joint observation-action space. Therefore, the value of RSA can represent the search space of the MARL method in the joint observation-action space. Policy homogenization results in high RSA values, indicating that agents are more likely to select the same action. This leads to the solidification of the search space of MARL methods within high-RSA subspaces of the joint observation-action space during training. The solidification mechanism is crucial to both the effectiveness and limitations of parameter sharing. In tasks where the optimal policy aligns with the solidification subspace—such as in the SMAC tasks—parameter sharing enhances training stability and training efficiency. However, in tasks where the optimal policy lies outside the solidified subspace—such as in the GRF tasks—parameter sharing impedes exploration in these regions, ultimately hindering performance. These findings emphasize the context-dependent nature of parameter sharing, revealing that its effectiveness is not universally applicable across all tasks. The effectiveness of parameter sharing is highly dependent on the specific characteristics of the task, emphasizing the need for enhancing its applicability across tasks.

Building upon the theoretical foundation, we propose a novel framework called **R**evising **S**hared **P**olicy (**RSP**) to balance the strengths and limitations of parameter sharing in cooperative MARL. The core idea of RSP is to revise the shared policy through agent-dependent adjustments, thereby enabling the framework to adaptively determine the appropriate search subspace for each task. We provide a theoretical justification showing that RSP achieves adaptive search by selectively preserving meaningful policy homogenization while avoiding meaningless ones. Meaningful homogenization helps identify effective search subspaces, facilitating efficient exploration and improving task performance. In contrast, meaningless homogenization can restrict the learning process to suboptimal regions of the policy space, hindering exploration and degrading overall performance. By mitigating these negative effects, the RSP framework allows parameter sharing to dynamically adjust the search subspace as needed—preserving performance on tasks where it already excels and enhancing performance tasks where it underperforms. Importantly, RSP modifies only the internal structure of the shared policy network without altering its input-output interface, ensuring compatibility with a broad range of MARL methods that utilize parameter sharing. To empirically validate the effectiveness of RSP, we integrate it with two representative MARL methods, QMIX [6] and MAPPO [17], and evaluate performance across two prominent testbeds: GRF [11] and SMACv2 [18]. The experimental results demonstrate that RSP can dynamically adjust the search subspace according to the specific task, leading to significant performance improvements.

In summary, our key contributions are as follows: 1) we theoretically reveal that the effectiveness and limitations of parameter sharing comes from its solidification for the search space of MARL methods in the joint observation-action space. 2) We propose the RSP framework, grounded in this theoretical insight, to balance these strengths and limitations by revising the shared policy in an agent-specific manner. 3) We validate RSP by integrating it with two representative MARL methods and evaluating it in two key testbeds. The results show that RSP improves the applicability of parameter sharing across tasks.

II. Preliminary

In this section, we first describe the problem formulation. Second, we introduce the observation similarity of commonly used tasks. Then, we introduce the definition of RSA and its division of joint observation-action. Finally, we present the proof that shared policy between agents would lead to policy homogenization.

A. Problem Formulation

We focus on fully cooperative multi-agent tasks where agents' observations are partial and the action space is discrete. These tasks are typically formulated as a decentralized partially observable Markov decision process (Dec-POMDP) [19]. Formally, Dec-POMDP is defined as a tuple $\mathcal{G} =< S, U, P, r, Z, n, \gamma >$. $s \in S$ describes the true states of the environments. At each timestep, $a \in A \equiv \{1, \cdots, n\}$ chooses an action $u^a \in U$, forming a joint action $\mathbf{u} \in \mathbf{U} \equiv U^n$. This causes a transition of the environment according to the state transition function $P(s'|s, \mathbf{u}) : S \times \mathbf{U} \times S \to [0, 1]$. All agents share the same team reward function $r(s, \mathbf{u}) : S \times \mathbf{U} \to \mathbb{R}$ and $\gamma \in [0, 1]$ is the discount factor. We consider a partially observable scenario in which each agent draws individual observations$z \in Z$ according to observation function $O(s, a) : S \times A \to Z$. Each agent has an action-observation history $\tau^a \in T \equiv (Z \times U)^*$, on which it conditions a stochastic policy $\pi^a(u^a|\tau^a) : T \times U \to [0, 1]$. The joint policy π has a joint action-value function $Q^\pi(s_t, \mathbf{u}_t) = \mathbb{E}_{s_{t+1:\infty}, \mathbf{u}_{t+1:\infty}}[R_t|s_t, \mathbf{u}_t]$, where $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ is the discounted return.

B. Observations Similarity

A key prerequisite for policy homogenization in cooperative MARL is the similarity of observations among agents. To

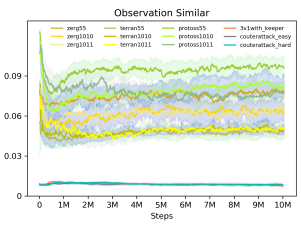


Fig. 1. Observations similarity on SMACv2 and GRF tasks.

investigate this, we conduct a detailed analysis of **observation similarity** in two widely used cooperative MARL benchmarks: SMACv2 [18] and GRF [11]. We introduce a metric to quantify **observation similarity**, which involves a two-step process. First, for each component of the observation at timestep t, we compute the standard deviation across all agents. Then, we calculate the mean of these standard deviations. This aggregated value provides a measure of **observation similarity** at timestep t: lower values indicate higher similarity in agent observations. Formally, let the observation of agent i at timestep t be denoted as $o_i^t = [o_{i1}^t, \cdots, o_{iL}^t]$, where L is the observation dimensionality and n is the number of agents. The matrix of observations for all agents at timestep t is:

$$C^t = \begin{bmatrix} o_{11}^t & \cdots & o_{1L}^t \\ \vdots & \ddots & \vdots \\ o_{n1}^t & \cdots & o_{nL}^t \end{bmatrix}. \tag{1}$$

For each column $j \in \{1, \ldots, L\}$, we define the vector $c_j^t = [o_{1j}^t, \cdots, o_{nj}^t]^{\mathrm{T}}$, which captures the j-th component of observations across agents. The standard deviation across agents is then computed for each dimension of observation, yielding the vector $d^t = [\mathrm{std}(c_1^t), \cdots, \mathrm{std}(c_L^t)]$, where $\mathrm{std}(\cdot)$ denotes the standard deviation. To extend the measure over training data, we average these values across the batch dimension. Specifically, the **observation similarity** for a batch of size B at timestep t is defined as:

$$os^t = \frac{1}{B} \sum_{b=1}^{B} d_b^t, \tag{2}$$

where d_b^t denotes the standard deviation vector for the b-th sample in the batch. This metric reflects the overall variation in observation components across agents. A lower value of os^t indicates that the agents' observations are more similar, which can facilitate policy homogenization due to the shared input features. In Fig. 1, we present the **observation similarity** of QMIX during training across a range of tasks from both SMACv2 and GRF. Our findings show that **observation similarity** in SMACv2 tasks is generally higher than in GRF tasks, though both environments exhibit consistently high similarity levels. This suggests that high **observation similarity** is a common feature in these tasks, underscoring the importance of addressing policy homogenization in cooperative MARL.

C. Rate of Same Action

In this section, we first introduce the definition of RSA. Secondly, we reveal how RSA divides the joint observation-action space into a set of subspaces. Finally, we prove two properties of this division.

RSA is defined as the highest proportion of same action in a joint action $\mathbf{u}^t = \{u_a^t\}_{a \in A}$ at timestep t. Mathematically, RSA is expressed as:

$$J_m^t = \frac{m}{n}, \quad 1 \leq m \leq n, m \in \mathbb{N}^+, \tag{3}$$

where m represents the largest amount of agents taking the same action in a joint action, and $\mathbb{N}^+$ represents the set of positive integers. The value of RSA is dependent on the number of agents and ranges from $1/n$ to 1, with a total of n possible values. Importantly, every joint action corresponds to a RSA value. As a result, the set for all joint actions can be divided into n subsets, with each subset corresponding to a specific RSA value. Considering that the joint observation-action space can be conceptualized as a two-dimensional space, where one dimension represents joint observations and the other represents joint actions. We can divide the joint observation-action space along the joint action dimension by RSA value. Thus, the joint observation-action space is divided into n subspaces, each corresponding to a distinct RSA value.

We demonstrate that the division of the joint observation-action space according to RSA is both complete and mutually exclusive. The completeness of the subspace partition provides a theoretical foundation for subsequent analyses, ensuring that the entire joint observation-action space is captured by the subspaces defined by RSA. Additionally, the mutual exclusivity of these subspaces guarantees that variations in RSA can effectively reflect changes in the search subspace. For convenience, we denote the joint observation-action space as $\mathcal{K}$.

Theorem 1. *The division of the joint observation-action space according to RSA is complete, i.e.,* $\bigcup \mathcal{K}_i = \mathcal{K}, 1 \leq i \leq n, i \in N^+$.

Proof. It is clear that there is a one-to-one correspondence between each subspace and the RSA value after division. Specifically, subspace $\mathcal{K}_i$ corresponds to RSA value $J_i^t = i/n$, where $1 \leq i \leq n$ and $i \in N^+$. Since all RSA values are distinct and complete, the division of the joint observation-action space is complete. Therefore, we have $\bigcup \mathcal{K}_i = \mathcal{K}$. □

Theorem 2. *The divided subspaces* $\mathcal{K}_i$ *is mutual exclusive, i.e.,* $\mathcal{K}_i \bigcap \mathcal{K}_j = \emptyset, i \neq j$.

Proof. Suppose there exists a joint action $\boldsymbol{u}$ that belongs to both subspaces $\mathcal{K}_i$ and $\mathcal{K}_j$, where $i \neq j$. Then, the RSA value corresponding to subspace $\mathcal{K}_i$ is $J_i^t = i/n$, and the RSA value corresponding to subspace $\mathcal{K}_j$ is $J_j^t = j/n$. According to the definition of RSA, it must hold that $J_i^t = i/n = J_j^t = j/n$, which leads to a contradiction because of $i \neq j$. Therefore, a joint action cannot belong to both subspaces simultaneously, and we conclude that $\mathcal{K}_i \bigcap \mathcal{K}_j = \emptyset, i \neq j$. □

In summary, RSA serves two critical functions: it divides the joint observation-action space into distinct subspaces and enables us to track which subspace the MARL method occupies during training or testing. This division is essential for understanding and monitoring the dynamics of exploration within MARL methods.

D. Policy Homogenization

To demonstrate that parameter sharing leads to policy homogenization and to establish a theoretical foundation for the RSP framework, we first introduce a suitable representation of the policy network under parameter sharing. This representation must satisfy two essential requirements: 1) **Capability to represent all possible shared policies**: In MARL, each agent's policy can be viewed as a function that maps observations to actions. Similarly, a shared policy network can be regarded as a function from observations to actions. Therefore, any function that takes an observation as input and produces an action as output should be representable within the shared policy framework. 2) **Ability to associate individual agent policies with the shared policy**: When adopting parameter sharing, the learning objective shifts from training n separate networks (each representing an agent's policy) to learning n functions through a single shared network. This transformation complicates the identification and analysis of individual agent contributions within the shared policy structure. Inspired by the mixture model framework [20], [21], we propose the following shared policy representation:

$$\textbf{Policy}(o_a) = \sum_{k=1}^{n} \pi_k f_k(o_a), \quad a \in A, \tag{4}$$

where o_a denotes the observation of agent a, f_k represents the k-th component function, and π_k is the corresponding mixing coefficient, subject to $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{n} \pi_k = 1$. This formulation satisfies the first requirement by allowing the component functions f_k to be unconstrained, thus capable of representing any possible policy. It also satisfies the second requirement by setting the number of component functions equal to the number of agents, enabling a potential one-to-one correspondence between component functions and individual agent policies. This structure provides a clear and interpretable foundation for analyzing shared policy representations in MARL.

Building upon the proposed representation of the shared policy, we proceed to demonstrate that policy homogenization is an inherent consequence of parameter sharing.

Theorem 3. *Shared policy tends to choose the same action when observations from different agents are similar.*

Proof. To prove this statement, we verify that two agents with shared policy would choose the same action under similar observations. We assume that at timestep t, agent i and agent j ($i, j \in A$) receive local observations o_i^t and o_j^t respectively. The difference between the policies of two agents is

$$\begin{aligned} \textbf{Policy}(o_i^t) - \textbf{Policy}(o_j^t) = & \overbrace{\sum_{k=1}^{n} [\pi_k (f_k(o_i^t) - f_k(o_j^t))]}^{(1)} \\ & + \overbrace{\sum_{k=1}^{n} [(\pi_k - \pi_k') f_k(o_j^t)]}^{(2)} \end{aligned} \tag{5}$$

For stable training, the shared policy network needs to be stable to input disturbances, i.e., it should obey the Lipschitz Continuity constraint. This constraint holds for the component of the shared policy network due to linearity. When the observations of two agents, o_i^t and o_j^t, are similar, they can be assumed to be any two observations in the δ-neighborhood of observation o^t, i.e., $o_i^t, o_j^t \in [o^t - \delta, o^t + \delta]$. Expanding $f_k(o_i^t)$ and $f_k(o_j^t)$ respectively at o^t with Taylor expansion (For simplicity, we only present first-order Taylor, while higher-order can get the same conclusion):

$$\begin{aligned} f_k(o_i^t) - f_k(o_j^t) &= [f_k(o^t) + f_k'(o^t)(o_i^t - o^t)] \\ &\quad - [f_k(o^t) + f_k'(o^t)(o_j^t - o^t)] \\ &= f_k'(o^t)(o_i^t - o_j^t). \end{aligned} \tag{6}$$

Therefore, $\lim_{o_i^t \to o_j^t} \sum_{k=1}^{n} [\pi_k (f_k(o_i^t) - f_k(o_j^t))] = \lim_{o_i^t \to o_j^t} \sum_{k=1}^{n} [\pi_k f_k'(o^t)(o_i^t - o_j^t)] \to 0$. As a result, the term (1) of Eq. 5 will converge to 0 when the observations are similar. For the term (2), its value depends on the mixing coefficient. If the mixing coefficients depend on observations of agents(we refer this function as $c(o_t)$, then $\pi_k = c(o_i^t)$ and $\pi_k' = c(o_j^t)$), we have $\lim_{o_i^t \to o_j^t} \sum_{k=1}^{n} [(c(o_i^t) - c(o_j^t)) f_k(o_j^t)]$. Similarly, the function c is also fitted by the neural network and also obeys the Lipschitz continuity constraint. Then, this term equal to $\lim_{o_i^t \to o_j^t} \sum_{k=1}^{n} [c'(o^t) f_k(o_j^t)(o_i^t - o_j^t)] \to 0$. If the mixing coefficients are independent of the observations of agents, then they should be constant values, which means $\pi_k = \pi_k'$. Then this term also converges to 0. Therefore, we can conclude that the term (2) of Eq. 5 converges to 0 when the local observations are similar. In summary, when the observations are similar, the difference between the policies of these two agents tends to be zero.

From the output of the policy network to the selection of action also requires a decision-making process. After analyzing the policy difference between two agents with similar observations, we need to examine the decision-making process for selecting actions. Despite differences in the decision-making process between value-based and policy-based methods, both methods select actions based on the highest component in the output vector of the policy network during execution. Consequently, when the output difference between the policy networks of two agents is close to zero, the highest components become identical, leading to agents selecting the same

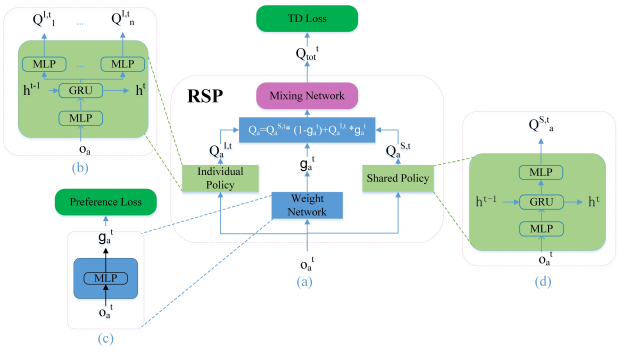


Fig. 2. (a) The implementation of RSP based on QMIX. (b) The details of individual policy networks. (c) The details of weight network. (d) The details of shared policy network.

action. Thus, in the case of similar observations, two agents will choose the same action. □

III. Method

In this section, we first introduce the construction process of RSP. Then, we prove that RSP-revised shared policies can avoid meaningless policy homogenization. Finally, we take QMIX as an example to introduce the implementation of RSP.

A. Revision Shared Policy

We have established that the shared policy distinguishes each agent based on their observations. However, when the observations of multiple agents exhibit similarity, the shared policy loses its ability to differentiate among individual agents. To mitigate this issue, revising the shared policy becomes necessary. An intuitive approach is introducing an individual policy component for each agent as a revision value. Consequently, the policy of agent i becomes

$$\textbf{Policy}_a(o_a) = \textbf{Policy}^S(o_a) + \textbf{Policy}^I_a(o_a) \tag{7}$$

where $\textbf{Policy}^S$ is the shared policy, and $\textbf{Policy}^I_a$ represents the individual policy component of agent $a \in A$ and takes effect when the observation for agent o_a is encountered. Eq.7 treats shared and individual policies equally, but in fact, their roles should dynamically adjust with varying observations. To achieve this, we introduce an observation-based weight function to dynamically blend these two components:

$$\textbf{Policy}_a(o_a) = (1 - g(o_a)) * \textbf{Policy}^S(o_a) + g(o_a) * \textbf{Policy}^I_i(o_a) \tag{8}$$

where $g(\cdot)$ represents the observation-based weight function and $g(o_a) \in [0, 1]$. In summary, RSP revises the form of shared policy so that it has the property of avoiding meaningless policy homogenization, which we will prove in the next subsection.

B. Avoid meaningless Policy Homogenization

In this subsection, we demonstrate that RSP can avoid meaningless policy homogenization. The difference between the revised policies of the two agents is

$$\textbf{Policy}(o_i^t) - \textbf{Policy}(o_j^t) = \overbrace{\sum_{k=1}^{n} [\pi_k f_k(o_i^t) - \pi'_k f_k(o_i^t)]}^{(1)} + \overbrace{g_j * \sum_{k=1}^{n} \pi'_k f_k(o_j^t) - g_i * \sum_{k=1}^{n} \pi_k f_k(o_i^t)}^{(2)} + \overbrace{g_i * f_i(o_i^t) - g_j * f_j(o_j^t)}^{(3)} \tag{9}$$

The term (1) converges to 0 which has been proved. We extend term (2) as follow:

$$g_j * \sum_{k=1}^{n} \pi'_k f_k(o_j^t) - g_i * \sum_{k=1}^{n} \pi_k f_k(o_i^t) = \overbrace{g_i * \sum_{k=1}^{n} [\pi'_k f_k(o_j^t) - \pi_k f_k(o_i^t)]}^{(2.1)} + \overbrace{(g_j - g_i) * \sum_{k=1}^{n} \pi'_k f_k(o_j^t)}^{(2.2)} \tag{10}$$

Similar to the previous case, the term (2.1) converges to 0. Different from previous cases, whether the term (2.2) converges to 0 depends on the output of the weight network. Given that the weight network is a function of observations and satisfies the Lipschitz Continuity constraint, we can get $\lim_{o_i^t \to o_j^t} (g_j - g_i) \to 0$, which results in the term (2.2) converging to 0. Thus, the term (2) converges to 0. We extend term (3) as follow:

$$g_i * f_i(o_i^t) - g_j * f_j(o_j^t) = \underbrace{g_j * (f_i(o_i^t) - f_j(o_j^t))}_{(3.1)} + \underbrace{(g_i - g_j) * f_i(o_i^t)}_{(3.2)} \tag{11}$$

Similar to the term (2.2), the term (3.2) also converges to 0. Different from other terms, whether the term (3.1) converges to 0 depends on the individual policies f_i and f_j. The individual policy network is trained solely on its trajectories and has no access to other agents' trajectories. This eliminates the influence of other agents' policies on the individual policy. If the individual policies of two agents are similar, the revised policies of these agents will also be similar when their local observations are similar. This phenomenon aligns with our intuition. If the individual policies of two agents are dissimilar, the revised policies of these agents will also be dissimilar even if their local observations are similar. This phenomenon is desirable as it avoids meaningless policy homogenization.

C. The Implementation of RSP

Obviously, RSP focuses solely on the the policy network and can be integrated with various existing MARL methods. We take QMIX as an example to show the implementation of RSP in Fig. 2. The combination with other MARL methods are similar. The implementation of the RSP framework involves

three key components: individual policy, shared policy, and observation-based weight. For the individual policy, n distinct policy networks are created for each agent. Recognizing that all agents should possess consistent knowledge about the task, the lower layers of each agent's individual policy network are shared, as depicted in Fig. 2(b). This design significantly reduces the number of additional parameters introduced and ensures that the feature representations across individual policy networks are consistent and stable. For the shared policy implementation, we keep the classic QMIX implementation, and the details are shown in Fig. 2(d). To implement observation-based weight, we introduce a weight network, whose structure is illustrated in Fig. 2(c). Since policy homogenization is closely tied to the similarity of observations, the weight network's preference is guided by **observation similarity**. Specifically, when observations are more similar, the weight network should favor the individual policy, thereby preserving diversity in the agents' behaviors. We introduce a preference loss to implement this concept.

$$\mathcal{L}_{pre} = \sum_{a \in A} |g_a^t - \mathcal{T}^t|, \quad \mathcal{T}^t = \textit{smooth\_step}(os^t). \tag{12}$$

The smooth_step [22] is a function similar to the sigmoid function and can output exact zeros and ones, thus allowing for completely preference. The total loss function for CSP combined with QMIX is $\mathcal{L} = \mathcal{L}_{QMIX} + \beta \mathcal{L}_{pre}$. $\mathcal{L}_{QMIX}$ denotes the loss function of QMIX, and β is a hyperparameter controlling the trade-off between the loss function of QMIX and the preference loss.

IV. Experiments

In this section, we present the experiment setup employed in our study. we then evaluate the performance of RSP in different environments. Finally, we conduct an ablation study to investigate the impact of each component of RSP on performance.

A. Experimental Setting

Environments We conduct experiments to evaluate the performance of RSP on two benchmark environments: GRF [11] and SMACv2 [18]. SMAC [9] is a popular benchmark for cooperative MARL and SMACv2 incorporates more randomness to the environment, making it more challenging. RSP's performance is assessed across all nine SMACv2 tasks. GRF serves as a contrasting environment where parameter sharing fails.

Baselines and Evaluation Metric We implement RSP with two representational cooperative MARL methods(fine-tuned QMIX [23], referred to as QMIX and MAPPO [17]) to evaluate the improvement of RSP on performances. To ensure a fair comparison, the hyperparameters remain consistent pre and post combination. Specifically, we conduct 32 test episodes without exploration and report the mean and variance of the test winning rate across five distinct seeds.

B. Performances and Adaptive Search

We present the RSA curves of QMIX, MAPPO, and their RSP implementations on the SMACv2 tasks in Fig. 3(a1-a3). The RSA curves for both QMIX and MAPPO across the nine SMACv2 tasks exhibit two notable characteristics. First, these curves remain stable throughout the training process, indicating that parameter sharing solidifies the search space of MARL methods within the joint observation-action space. Second, the curves stabilize at higher RSA values, suggesting that policy homogenization persists during training. This phenomenon reflects the central role of parameter sharing in constraining the search space and promoting action similarity among agents. The RSA curves for the RSP implementations of both QMIX and MAPPO exhibit the same trends. The stability of the search space in RSP on the SMACv2 tasks aligns with the established understanding that parameter sharing has a positive impact on these tasks. Moreover, RSP enhances the performance of QMIX on 7 tasks, improves the performance of MAPPO on 5 tasks, and maintains the performance on the remaining tasks (as shown in Fig. 3(b1-b3), (c1-c3), and (d1-d3)). This demonstrates that RSP effectively adapts the search space within the joint observation-action space in SMACv2 tasks and mitigates the negative impact of policy homogenization through its structural design, leading to improved performance.

We now turn to the RSA curves of QMIX, MAPPO, and their RSP implementations on the GRF tasks, as shown in Fig. 4(a). The RSA curves for QMIX and MAPPO exhibit markedly different characteristics on the GRF tasks compared to the SMACv2 tasks. While the RSA curves remain relatively stable during training for some tasks, the fluctuations are significantly more pronounced. In fact, the RSA curves for two of the tasks display a downward trend early in training, eventually stabilizing in the later stages. This suggests that the diverse policies required for the GRF tasks prompt MARL methods to explore a larger search space, but policy homogenization hinders this exploration, limiting the effectiveness of parameter sharing. In contrast, the RSA curves for the RSP implementations of both QMIX and MAPPO on the GRF tasks show entirely different behavior. These curves remain stable throughout the training process and stabilize at lower RSA values, indicating that RSP successfully mitigates the negative effects of policy homogenization while preserving the ability of parameter sharing to solidify the search space of MARL methods. Furthermore, as illustrated in Fig. 4(b-d), RSP significantly improves the performance of both QMIX and MAPPO on the GRF tasks.

In conclusion, the RSP framework enables MARL methods that adopt parameter sharing to adaptively determine the search space based on the specific characteristics of each task. This adaptive capability enhances the cross-task applicability of parameter sharing, making it more applicable across a wide range of tasks.

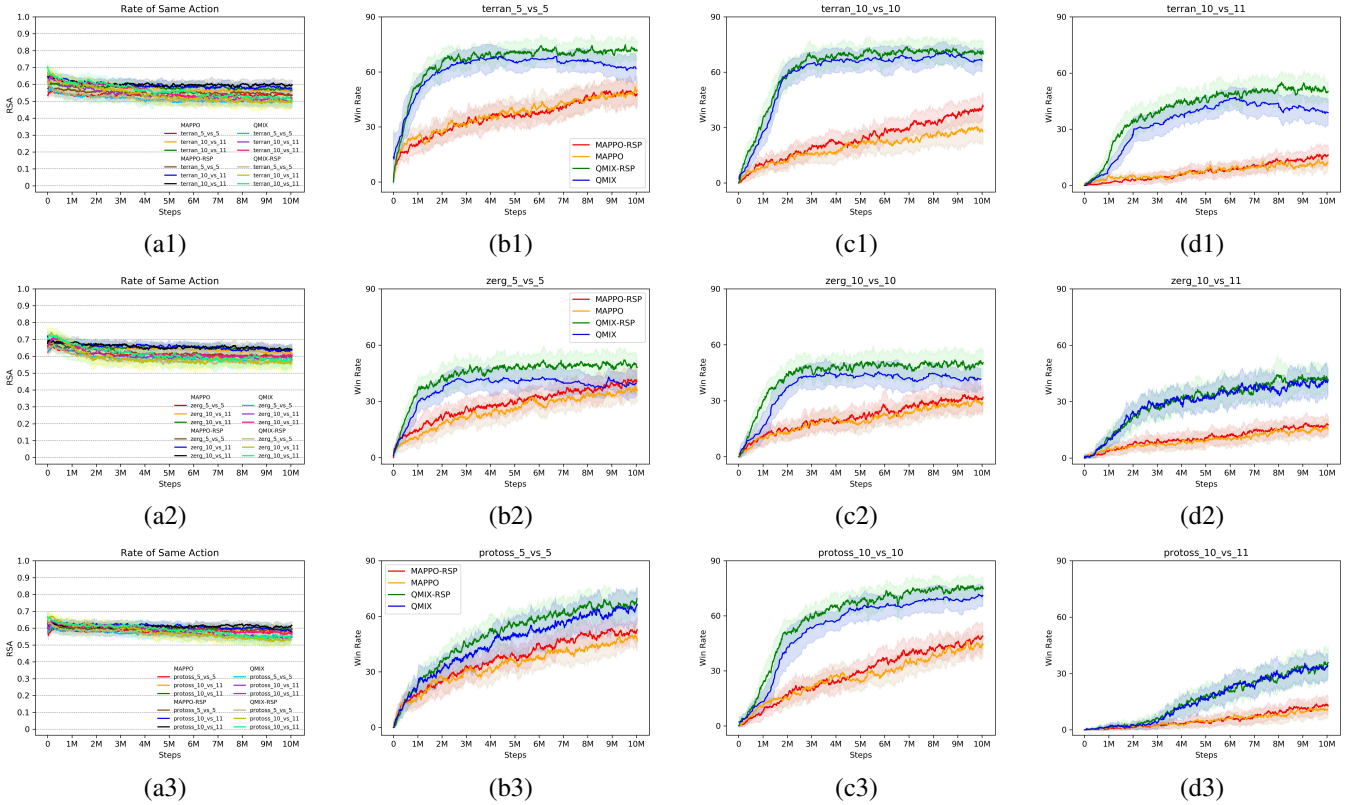


Fig. 3. Search spaces on SMACv2 tasks and the corresponding performances.

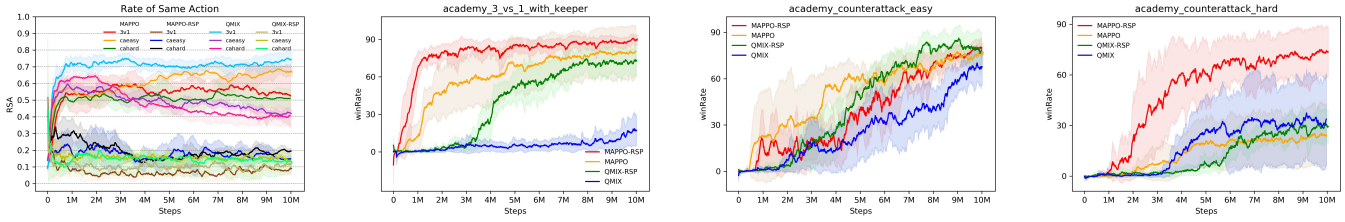
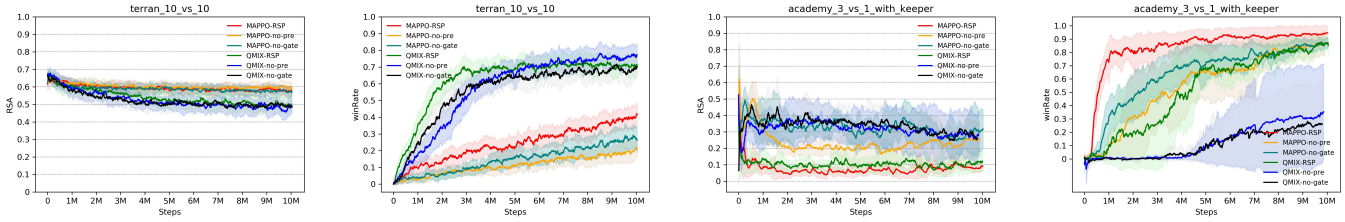


Fig. 4. Search sapces on GRF tasks and the corresponding performances. The -RSP indicates combined with RSP framework.

C. Ablation Study

To evaluate the individual contributions of the key components in RSP, we conduct ablation studies aimed at assessing the effectiveness of the preference loss and the weight network. Notably, the individual policy network was excluded from this study, given that its removal would essentially revert the MARL method to its original form. The evaluation metrics, encompassing test winning rates and corresponding RSA curves, were measured throughout training steps for the ablation variants, and the results are presented comprehensively in Fig. 5.

Ablating the preference loss and weight network on the SAMCv2 task has no effect on the search subspace because the search subspace of the SMACv2 task does not need to be changed. When the preference loss is ablated in QMIX-RSP, there is a noticeable decrease in convergence speed but an improvement in final performance. This occurs because the individual policy network introduces diversity in revising the shared policy, which complicates convergence but ultimately enhances performance. In contrast, the ablation of preference loss in MAPPO-RSP results in both slower convergence and reduced final performance on the SMACv2 task. Additionally, removing the weight network from the SMACv2 task leads to a significant decline in both convergence speed and final performance. When the preference loss and weight network are ablated in the GRF task, the method's ability to modify the search subspace is notably diminished, leading to substantial drops in both convergence speed and final performance. In summary, while the preference loss introduces some challenges in convergence, its overall benefits outweigh the drawbacks, making it a worthwhile trade-off. The weight network consistently plays a positive role in enhancing performance



(a) (b) (c) (d)

Fig. 5. Ablation study on SMACv2 and GRF.

and convergence. Both components are essential for RSP to improve the cross-task applicability of MARL methods.

V. Conclusion

In this work, we present a comprehensive theoretical analysis of parameter sharing, aiming to uncover the underlying mechanisms that contribute to its successes and failures. Our analysis reveals that the effectiveness of parameter sharing arises from its solidification of the search space in the joint observation-action space.Parameter sharing achieves this solidification through a key phenomenon called policy homogenization. This solidification mechanism, as outlined in our analysis, acts as both an enabler and a constraint on the effectiveness of parameter sharing, contingent on whether the optimal policy for a given task is retained or pruned. We also introduce RSA to reveal and observe this mechanism. Building on these theoretical insights, we introduce RSP to allow parameter sharing to adaptively determine the search subspace according to the specific characteristics of the task. We theoretically establish that RSP can selectively maintain meaningful policy homogenization while avoiding meaningless ones. By mitigating these negative effects, the RSP framework endow parameter sharing the ability to adjust the search subspace as required, thereby improving its applicability. We implement RSP into two representative MARL methods and conduct experiments in two well-established MARL testbeds. The experimental results confirm that RSP can dynamically adjust the search subspace based on tasks, leading to improved performance of the original methods. As a result, RSP enhance the applicable scope of parameter sharing.

References

[1] M. Zhou, J. Luo, J. Villella, Y. Yang, D. Rusu, J. Miao, W. Zhang, M. Alban, I. Fadakar, Z. Chen, *et al.*, "Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving," *arXiv preprint arXiv:2010.09776*, 2020.

[2] A. J. Singh, A. Kumar, and H. C. Lau, "Hierarchical multiagent reinforcement learning for maritime traffic management," 2020.

[3] J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente, *et al.*, "Pettingzoo: Gym for multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15032–15043, 2021.

[4] C. Li, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang, "Celebrating diversity in shared multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3991–4002, 2021.

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

[6] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International conference on machine learning*, pp. 4295–4304, PMLR, 2018.

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

[8] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "Qplex: Duplex dueling multi-agent q-learning," *arXiv preprint arXiv:2008.01062*, 2020.

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

[10] J. Jiang and Z. Lu, "The emergence of individuality," in *International Conference on Machine Learning*, pp. 4992–5001, PMLR, 2021.

[11] K. Kurach, A. Raichuk, P. Stańczyk, M. Zajac, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, *et al.*, "Google research football: A novel reinforcement learning environment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 4501–4510, 2020.

[12] M. Bettini, A. Shankar, and A. Prorok, "Heterogeneous multi-robot reinforcement learning," *arXiv preprint arXiv:2301.07137*, 2023.

[13] S. Hu, C. Xie, X. Liang, and X. Chang, "Policy diagnosis via measuring role diversity in cooperative multi-agent rl," in *International Conference on Machine Learning*, pp. 9041–9071, PMLR, 2022.

[14] T. Wang, H. Dong, V. Lesser, and C. Zhang, "Roma: Multi-agent reinforcement learning with emergent roles," *arXiv preprint arXiv:2003.08039*, 2020.

[15] T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson, and C. Zhang, "Rode: Learning roles to decompose multi-agent tasks," *arXiv preprint arXiv:2010.01523*, 2020.

[16] X. Yu, Y. Lin, X. Wang, S. Han, and K. Lv, "Ghq: Grouped hybrid q learning for heterogeneous cooperative multi-agent reinforcement learning," *arXiv preprint arXiv:2303.01070*, 2023.

[17] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24611–24624, 2022.

[18] B. Ellis, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. N. Foerster, and S. Whiteson, "Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning," *arXiv preprint arXiv:2212.07489*, 2022.

[19] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*. Springer, 2016.

[20] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*, vol. 38. M. Dekker New York, 1988.

[21] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite mixture models," *Annual review of statistics and its application*, vol. 6, pp. 355–378, 2019.

[22] H. Hazimeh, N. Ponomareva, P. Mol, Z. Tan, and R. Mazumder, "The tree ensemble layer: Differentiability meets conditional computation," in *International Conference on Machine Learning*, pp. 4138–4148, PMLR, 2020.

[23] J. Hu, S. Jiang, S. A. Harding, H. Wu, and S.-w. Liao, "Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning," *arXiv e-prints*, pp. arXiv–2102, 2021.