GHQ: Grouped Hybrid Q Learning for Heterogeneous Cooperative Multi-agent Reinforcement Learning

Xiaoyang Yu[1] , **Youfang Lin**[1] , **Xiangsen Wang**[1] , **Sheng Han**[1] , **Kai Lv**[1]
[1]Beijing Jiaotong University
{lvkai, 19112039}@bjtu.edu.cn

arXiv:2303.01070v1 [cs.MA] 2 Mar 2023

Abstract

Previous deep multi-agent reinforcement learning (MARL) algorithms have achieved impressive results, typically in homogeneous scenarios. However, heterogeneous scenarios are also very common and usually harder to solve. In this paper, we mainly discuss cooperative heterogeneous MARL problems in Starcraft Multi-Agent Challenges (SMAC) environment. We firstly define and describe the heterogeneous problems in SMAC. In order to comprehensively reveal and study the problem, we make new maps added to the original SMAC maps. We find that baseline algorithms fail to perform well in those heterogeneous maps. To address this issue, we propose the Grouped Individual-Global-Max Consistency (GIGM) and a novel MARL algorithm, Grouped Hybrid Q Learning (GHQ). GHQ separates agents into several groups and keeps individual parameters for each group, along with a novel hybrid structure for factorization. To enhance coordination between groups, we maximize the Inter-group Mutual Information (IGMI) between groups' trajectories. Experiments on original and new heterogeneous maps show the fabulous performance of GHQ compared to other state-of-the-art algorithms.

1 Introduction

Previous multi-agent reinforcement learning (MARL) methods have achieved impressive results in cooperative environment [Hernandez-Leal *et al.*, 2019; Gronauer and Diepold, 2022], *e.g.* the Starcraft Multi-Agent Challenges (SMAC) [Samvelyan *et al.*, 2019]. The original SMAC maps mainly consist of homogeneous maps (Appendix 1). However, real scenarios also contain many heterogeneous problems, such as wireless network accessibility problem [Yu *et al.*, 2021b] and multi-agent robotic systems [Ivić, 2020; Yoon *et al.*, 2019]. Thus, it is beneficial to enrich the SMAC environment with more heterogeneous maps.

Recent approaches solve the heterogeneous problem with various individual agent policies [Kuba *et al.*, 2021; Bono *et al.*, 2018]. However, these approaches lack the definition and further analysis of the property of heterogeneous problem. In this paper, we define the *heterogeneous MARL* problem as the agents within the multi-agent system (MAS) possess diverse properties. Specifically, in this heterogeneous scenario, the transition tuples $(\boldsymbol{o}^t, s^t, \boldsymbol{a}^t, r^t)$ of distinct agent types are different, leading to remarkably different policies. Considering the generation process of transition tuple, we conclude that the heterogeneity in MARL mainly occur in three components of the tuple: *Local Reward*, *Local Observation*, and *Local Transition*. The default reward function $R(s, \boldsymbol{a})$ in SMAC is global and universal among all agents, so the *Local Reward Heterogeneity* does not exist. The observation function $O(s^t, i)$ in SMAC is also global, generating local observations o_i^t with same components. So the *Local Observation Heterogeneity* does not exist either. In this paper, we focus on the *Local Transition Heterogeneity* (LTH).

After analyzing the original SMAC maps, we find that most of the original heterogeneous maps are *symmetric*. However, the enemies controlled by the internal AI script are unable to utilize the heterogeneity. As a result, these symmetric heterogeneous maps are not as hard as they are expected to be. We provide experimental analysis to demonstrate that symmetric map can help MARL agents acquire advantage against the internal AI script. Therefore, in order to fully reveal LTH, we design new asymmetric heterogeneous maps and test baseline algorithms. As our expectation, baseline methods fail to achieve high winning rate in our new maps.

A natural solution for LTH is **grouping**. Specifically, an agent is determined to a specific group depending on its transition property (*e.g.* action-dim or speed). First, we need to generalize the Individual-Global Maximum (IGM) consistency [Son *et al.*, 2019] into grouped situation. Therefore, we conduct the Grouped Individual-Global Maximum (GIGM) consistency and a condition to test whether a grouping method satisfy GIGM. Next, we propose the Grouped Hybrid Q Learning (GHQ). Agents are partitioned into groups with same transition property before training. Each group has its own network and only shares parameter within the group. We further propose a novel hybrid structure for factorization. For better cooperation, a variational lower bound of mutual information (MI) is introduced to increase correlation between groups. Finally, we test GHQ in our new asymmetric heterogeneous maps. Results show that GHQ outperforms other baseline methods, and the cooperate policy between GHQ groups is different against baselines.

2 Related Works

Following the centralized training with decentralized execution (CTDE) paradigm [Foerster *et al.*, 2016; Kraemer and Banerjee, 2016; Gupta *et al.*, 2017], which requests agents not to use state s during execution, recent approaches have achieved impressive results in SMAC environment. The mainstream value-based method is the value decomposition method, of which the formal objective is to learn a centralized yet factorized joint action-value function Q_{tot} and the factorization structure: $Q_{tot} \rightarrow Q_i$. In order to factorize Q_{tot} and use the greedy policy of Q_i to select actions, the Individual-Global-Max (IGM) consistency [Son *et al.*, 2019] is required:

$$\operatorname{argmax} Q_{tot}(\boldsymbol{\tau}, s) = \begin{pmatrix} \operatorname{argmax} Q_1(\tau_1) \\ ..., \\ \operatorname{argmax} Q_k(\tau_k) \end{pmatrix} . \quad (1)$$

VDN [Sunehag *et al.*, 2017] represents Q_{tot} as the sum of local Q_i functions. QMIX [Rashid *et al.*, 2018] changes the factorization structure from **additivity** to **monotonicity**, and the fine-tuned version of QMIX has been proved to be one of the best methods on the original SMAC maps [Hu *et al.*, 2021]. Based on these two fundamental methods, QTRAN [Son *et al.*, 2019], WQMIX [Rashid *et al.*, 2020a], Qatten [Yang *et al.*, 2020], and QPLEX [Wang *et al.*, 2020a] improve performance with modified value factorizing mechanism.

Heterogeneous MARL has been considered as a special case of homogeneous MARL and can be handled with individual policy networks. HAPPO [Kuba *et al.*, 2021], in which the H stands for heterogeneous, lacks specific analysis and sufficient experiments for heterogeneity. In other field of MAS, [Yang and Parasuraman, 2021] uses Relative Needs Entropy (RNE) to build a trust model to improve cooperation in heterogeneous multi-robot grouping task, and [Hartmann *et al.*, 2021] contributes a novel method for the heterogeneous multi-robot assembly planning.

Grouping is a natural solution for complex or large scale problems and is widely used in many problems of machine learning. [Rotman *et al.*, 2020] enhances the Optimal Sequential Grouping (OSG) to solve the video scene detection problem. [Ling *et al.*, 2022] proposes FedEntropy for better dynamic device grouping in federated learning. [Hou *et al.*, 2022] introduces an enhanced decentralized autonomous aerial swarm system with group planning. [Al Faiya *et al.*, 2021] designs a self-organizing MAS for distributed voltage regulation in smart power grid.

Computing variational bound of mutual information (MI) has been proved to enhance cooperation in MARL. MAVEN [Mahajan *et al.*, 2019] maximizes a variational lower bound of the MI between the latent variable z and the agent-specific Boltzmann policy $\sigma(\tau)$ to encourage exploration of MARL algorithm. ROMA [Wang *et al.*, 2020b] computes two MI-related losses to learn both identifiable and specialized role policies. PMIC [Li *et al.*, 2022] maintains a positive and a negative trajectory memories to compute the upper bound and lower bound of the MI between global state s and joint action $\boldsymbol{a}$. MAIC [Yuan *et al.*, 2022] maximizes the MI between the trajectory of agent i and the ID of agent j for teammate modeling and communication. CDS [Li *et al.*, 2021] maximizes the MI between the trajectory of agent i and its own agent ID i to maintain diverse individual local Q functions.

3 Preliminaries

In this paper, we study the cooperative MARL problems that can be modeled as the decentralized partially observable Markov decision process (Dec-POMDP) [Oliehoek and Amato, 2016]. The problem is described with a tuple $G = \langle S, \boldsymbol{A}, P, R, \boldsymbol{\Omega}, O; \gamma, K, T \rangle$. $s \in S$ denotes the true state of environment with complete information, $K = \{1, ..., k\}$ denotes the finite set of k agents, and $\gamma \in [0, 1)$ is the discount factor. At each time-step $t \leq T$, agent $i \in K$ receives an individual *partial observation* o_i^t and chooses an action $a_i^t \in A_i$ from local action set A_i, with the local action-dim $|A_i|$. All agents' actions form a joint action $\boldsymbol{a}^t = (a_1^t, ..., a_k^t) \in \boldsymbol{A} = (A_1, ..., A_k)$. The environment receives a joint action $\boldsymbol{a}^t$ and returns a next-state s^{t+1} according to the transition function $P(s^{t+1}|s^t, \boldsymbol{a})$, and a reward $r^t = R(s, \boldsymbol{a}^t)$ shared by all agents. The joint observation $\boldsymbol{o}^t = (o_1^t, ..., o_k^t) \in \boldsymbol{\Omega}$ is generated according to the observation function $O(s^t, i)$. Observation-action trajectory $\tau^t = \cup_0^t \{(o^t, a^t)\}$ is the summary of partial transition tuples before t. Replay buffer $\mathcal{D} = \cup(\boldsymbol{\tau}, s, r)$ stores all data for batch sampling. Network parameters are notated by θ and ψ.

4 Local Transition Heterogeneity

Local Transition Heterogeneity (LTH) means that agents choosing the same available action cannot reach the same next-state s^{t+1} from a same state s^t.

(Definition 1, Local Transition Heterogeneity (LTH)) Let there be agent i and j, with policies $\pi_i(a_i|s)$ and $\pi_j(a_j|s)$, transition functions $P_i(s^{t+1}|s^t, a_i)$ and $P_j(s^{t+1}|s^t, a_j)$, and a starting state s^t capable for all actions to be applied on. The sets of next-states $\{s_i^{t+1}|s^t, \pi_i, P_i\}$ and $\{s_j^{t+1}|s^t, \pi_j, P_j\}$ are generated by the two agents' policies individually executed on s^t. If the intersection of the two sets is empty for all available policies, then the MARL problem has LTH:

$$\{s_i^{t+1}|s^t, \pi_i, P_i\} \cap \{s_j^{t+1}|s^t, \pi_j, P_j\} = \emptyset, \ (\forall \pi_i, \pi_j) \ . \quad (2)$$

We divide the actions $\boldsymbol{A}$ in SMAC into two types, *common actions* A_{com} for moving and stopping, and *interactive actions* A_{act} for interacting with other units. We further conclude that differences in local action functionality and local action dynamics lead to the LTH with different transition function properties, corresponding to two sub-classes of LTH:

- *Local Functionality Heterogeneity* (LFH) means that agents are professionalized to finish specific tasks. Their transition functions are completely different in formula, and their next-states of individual policies s_i^{t+1} and s_j^{t+1} can never be the same state. For example, there are two agent types in SMAC, supporting units (U_{spt}) and attacking units (U_{atk}). U_{spt} can only affect allies while U_{atk} can only affect enemies. For instance, Medivac is a U_{spt} who can only heal allies, while Marine is a U_{atk} who can only attack enemies. Their objects and effects of A_{act} are different and therefore LFH occurs. In general, the difference in action objects leads to the

difference in action-dim $|A_i|$, and is sufficient for LFH. Therefore, the difference in $|A_i|$ can be used to prove the existence of LFH.

- *Local Dynamic Heterogeneity* (LDH) means that the dynamics of agents are different. Their transition functions are identical in formula but different in parameters. For example, in SMAC, Medivac is a flying unit while Marine is a ground unit. Their moving speed are different, so their transition functions of A_{com} are different only in parameters, leading to LDH. Their one-step transitions are not equal, but multi-step transitions $s^t \rightarrow s^{t+k}$ can have a same terminal state s^{t+k} with different trajectories τ_i and τ_j. For instance, Medivac flys a curve to reach s^{t+k} while Marine walks directly to s^{t+k}.

The default setup of SMAC environment and previous algorithms ignore the LTH. SMAC increases the action-dim $|A_i|$ of U_{spt} up to the same number of U_{atk} with a padding vector and masks it when choosing actions, which covers up the LTH. Previous algorithms apply parameter sharing among all unit types, which prevents the MAS from learning better coordinating policy. In GHQ, all agents are set to use their true $|A_i|$, and parameter sharing is restricted between agents within the same group.

5 Method

5.1 Grouped Individual-Global-Max Consistency

As is shown in section 4, LTH does not change the reward function $R(s, \boldsymbol{a})$ or the available action mask. Therefore, any available joint action $\boldsymbol{a}$ is rewarded the same as it in homogeneous scenarios, and the optimal joint action $\boldsymbol{a}^*$ is not affected. As a result, the IGM consistency in LTH still holds and we can further generalize it to a "grouped" situation for solving LTH problems with grouping value factorization.

(Definition 2, Grouped IGM Consistency (GIGM)) Let there be $U = \{1, ..., u\}$, $(u < k)$ agent groups in total. An agent group $\mathcal{G}_m$ $(m \in U)$ consists of agents arbitrarily predefined. If the argmax operation performed on the joint function Q_{tot} yields the same result as a set of individual argmax operations performed on all group functions $Q_{\mathcal{G}_m}$ $(m \in U)$; and the argmax operation performed on each group function $Q_{\mathcal{G}_m}$ yields the same result as a set of individual argmax operations performed on the agent functions Q_i $(i \in \mathcal{G}_m)$, then GIGM holds true:

$$\begin{aligned} \operatorname{argmax} Q_{tot}(\boldsymbol{\tau}, s) &= \begin{pmatrix} \operatorname{argmax} Q_{\mathcal{G}_1}(\boldsymbol{\tau}_{\mathcal{G}_1}, s) \\ ..., \\ \operatorname{argmax} Q_{\mathcal{G}_u}(\boldsymbol{\tau}_{\mathcal{G}_u}, s) \end{pmatrix} \\ &= \begin{pmatrix} \operatorname{argmax} Q_1(\tau_1) \\ ..., \\ \operatorname{argmax} Q_k(\tau_k) \end{pmatrix}, \\ \operatorname{argmax} Q_{\mathcal{G}_m}(\boldsymbol{\tau}_{\mathcal{G}_m}, s) &= \begin{pmatrix} \operatorname{argmax} Q_i(\tau_i) \\ i \in \mathcal{G}_m \end{pmatrix}, \end{aligned} \tag{3}$$

where $\boldsymbol{\tau}_{\mathcal{G}_m} = \cup_{i \in \mathcal{G}_m} \{\tau_i\}$ is the group trajectory of $\mathcal{G}_m$, $\boldsymbol{\tau} = \cup_{i \in K} \{\tau_i\}$ is the global joint trajectory of all agents. Furthermore, we conclude a theorem sufficient to prove GIGM:

(Theorem 1, Joint Trajectory Condition (JTC)) GIGM holds true if the following two conditions are simultaneously satisfied:

(i) The global joint trajectory is equivalent to the union of all group trajectories.

$$\boldsymbol{\tau} = \cup_{i \in K} \{\tau_i\} = \cup_{m \in U} \{\boldsymbol{\tau}_{\mathcal{G}_m}\} . \tag{4}$$

(ii) The intersection of all group trajectories is empty.

$$\cap_{i \in K} \{\boldsymbol{\tau}_{\mathcal{G}_m}\} = \emptyset . \tag{5}$$

The first condition guarantees the transitivity of argmax operations, which are performed on Q functions defined on trajectories τ. The second condition guarantees the coexistence of argmax operations on all $Q_{\mathcal{G}_m}$, and the equivalence of argmax operations on all group and agent functions: $\operatorname{argmax}\ Q_{\mathcal{G}_m}\ (m \in U) = \operatorname{argmax}\ Q_i\ (i \in K)$.

5.2 Local Transition Grouping

In order to utilize GIGM in LTH, we propose the **Local Transition Grouping (LTG)**, which means partitioning agents into different groups by their different transition function properties. Our goal is to acquire a grouping function $g(i, \mathcal{G}_m)(i \in K,\ m \in U)$ for agent i and group $\mathcal{G}_m$:

$$g(i, \mathcal{G}_m) = \begin{cases} 1 & \text{if } i \in \mathcal{G}_m \\ 0 & \text{else} \end{cases} . \tag{6}$$

In this paper, we choose action-dim $|A_i|$ for partitioning. Each agent group $\mathcal{G}_m$ consists of agents with the same action-dim $|A_{\mathcal{G}_m}|$. *Parameter sharing* is only allowed between agents within the same group and only one universal agent network is kept for one group, which significantly reduces the number of agent networks from K to U. Maintaining a proper parameter sharing structure not only avoids redundant computing resources for individual agent networks, but can also increase intra-group cooperating via homophily [Dong *et al.*, 2021]. Moreover, it is obvious that LTG is a mapping function from agents to groups $g(i, \mathcal{G}_m) : (K \rightarrow U)$, because every agent's $|A_i|$ must be set during the initialization of the environment and one agent can only be assigned to one specific group with $|A_i| = |A_{\mathcal{G}_m}|$. Therefore, JTC is satisfied and thus GIGM holds true. Furthermore, LTG avoids unexpected interference towards agent policies from the agents belong to different groups, resulting in better cooperating policy.

5.3 Inter-group Mutual Information Loss

In order to enhance inter-group cooperation and relation, we maximize the inter-group mutual information (IGMI) between trajectories of different groups $\mathcal{G}_m$ and $\mathcal{G}_n$, written as $I(\boldsymbol{\tau}_{\mathcal{G}_m}; \boldsymbol{\tau}_{\mathcal{G}_n})$. Because IGMI can only be calculated between two distributions, we add a Gaussian distribution layer in the agent network of every group, marked as $l_{\mathcal{G}_m}$ and $l_{\mathcal{G}_n}$. For encoding trajectories, we use GRU[Cho *et al.*, 2014] as the encoder and hidden states $h_{\mathcal{G}_m}$ and $h_{\mathcal{G}_n}$ of GRU are the input of $l_{\mathcal{G}_m}$ and $l_{\mathcal{G}_n}$ separately.

$$\begin{aligned} h_{\mathcal{G}_m} &= GRU(\boldsymbol{\tau}_{\mathcal{G}_m}) = GRU(\cup_{i \in \mathcal{G}_m} \{\tau_i\}) , \\ l_{\mathcal{G}_m} &= Gaussian(h_{\mathcal{G}_m}) , \\ I(\boldsymbol{\tau}_{\mathcal{G}_m}; \boldsymbol{\tau}_{\mathcal{G}_n}) &= I(l_{\mathcal{G}_m}; l_{\mathcal{G}_n} | h_{\mathcal{G}_m}, h_{\mathcal{G}_n}) . \end{aligned} \tag{7}$$

We further conduct a variational lower bound of IGMI for easier calculation (Appendix 2), and the IGMI loss of $\mathcal{G}_m$ is:

$$\begin{aligned}&\mathcal{L}_{MI_m}(\tau_{\mathcal{G}_m};\tau_{\mathcal{G}_n}|\psi_{\mathcal{G}_m})\\&=\mathbb{E}_{\mathcal{D}}[D_{KL}(p(l_{\mathcal{G}_m})||q_{\psi_{\mathcal{G}_m}}(l_{\mathcal{G}_m}|l_{\mathcal{G}_n},h_{\mathcal{G}_m}))]\,,\end{aligned}\tag{8}$$

where $\mathbb{E}_{\mathcal{D}}$ means sampling a batch of tuples $(\boldsymbol{\tau},s,r)$ from replay buffer $\mathcal{D}$ and calculating expectation across the batch, $q_{\psi_{\mathcal{G}_m}}$ is an inference distribution of group $\mathcal{G}_m$ with parameter $\psi_{\mathcal{G}_m}$, and D_{KL} is the KL-divergence. This function is symmetric between groups and $\mathcal{L}_{MI_n}$ can be written by replacing m to n. Because we must keep $q_{\psi_{\mathcal{G}_m}}$ independent from $h_{\mathcal{G}_n}$, a mixed input of different groups is forbidden. Therefore, we keep individual inference network q_ψ for each group.

5.4 Grouped Hybrid Q Learning

An ordinary idea to calculate $Q_{\mathcal{G}_m}$ and Q_i is to design factorization structures for $Q_{tot}\rightarrow Q_{\mathcal{G}_m}$ and $Q_{\mathcal{G}_m}\rightarrow Q_i$. Let $\mathcal{C}_{\mathcal{G}_m}$ and $\mathcal{C}_i$ be the two factor function. Like IGM [Rashid *et al.*, 2020b], the monotonicity constraint is also sufficient for GIGM. Therefore, $\mathcal{C}_{\mathcal{G}_m}$ and $\mathcal{C}_i$ can be written as:

$$\frac{\partial Q_{tot}(\boldsymbol{\tau},s)}{\partial Q_{\mathcal{G}_m}(\boldsymbol{\tau}_{\mathcal{G}_m},s)}=\mathcal{C}_{\mathcal{G}_m}\geq 0,\ m\in U\,,\tag{9}$$

$$\frac{\partial Q_{\mathcal{G}_m}(\boldsymbol{\tau}_{\mathcal{G}_m},s)}{\partial Q_i(\tau_i)}=\mathcal{C}_i\geq 0,\ i\in\mathcal{G}_m\,.\tag{10}$$

Our key insight is that we do not need to calculate either $\mathcal{C}_{\mathcal{G}_m}$ or the true value of $Q_{\mathcal{G}_m}$. Instead of hierarchical factorization structure, we imply *independent Q Learning* (IQL) [Tan, 1993] for $\mathcal{C}_{\mathcal{G}_m}$, which makes $Q_{\mathcal{G}_m}$ become an action-value function instead of a utility function [Guestrin *et al.*, 2001; Rashid *et al.*, 2020b] and $\mathcal{C}_{\mathcal{G}_m}$ become a positive constant. This is our **hybrid** structure for factorization:

$$\begin{aligned}\mathcal{L}_{TD_m}(\boldsymbol{\theta}_{\mathcal{G}_m})&=\mathbb{E}_{\mathcal{D}}[(y^{\mathcal{G}_m}-Q_{\mathcal{G}_m}(\boldsymbol{\tau}_{\mathcal{G}_m},s;\boldsymbol{\theta}_{\mathcal{G}_m}))^2]\,,\\y^{\mathcal{G}_m}&=r+\gamma\max_{\boldsymbol{a}'}Q^{tgt}_{\mathcal{G}_m}(\boldsymbol{\tau}'_{\mathcal{G}_m},s';\boldsymbol{\theta}^{tgt}_{\mathcal{G}_m})\,,\end{aligned}\tag{11}$$

where $y^{\mathcal{G}_m}$ is the TD-target of $Q_{\mathcal{G}_m}$, $Q^{tgt}_{\mathcal{G}_m}$ is the target Q function of $Q_{\mathcal{G}_m}$, and $\boldsymbol{\theta}^{tgt}_{\mathcal{G}_m}$ and $\boldsymbol{\theta}_{\mathcal{G}_m}$ are network parameters of $Q^{tgt}_{\mathcal{G}_m}$ and $Q_{\mathcal{G}_m}$, separately. Group network $\boldsymbol{\theta}_{\mathcal{G}_m}$ consists of two parts, agent network θ_i and mixing network θ_{M_m}, whose parameters are shared among all agents of group $\mathcal{G}_m$ and receive their losses following (10):

$$\frac{\partial\mathcal{L}_{TD_m}(\boldsymbol{\theta}_{\mathcal{G}_m})}{\partial\theta_i}=\frac{\partial\mathcal{L}_{TD_m}(\boldsymbol{\theta}_{\mathcal{G}_m})}{\partial\theta_{M_m}}\cdot\frac{\partial\theta_{M_m}}{\partial\theta_i}\,.\tag{12}$$

GIGM and the input of state information s keep the different group value functions in relevant, and IGMI further enhances the relation. Even though IQL method suffers from non-stationary problem [Foerster *et al.*, 2017], GHQ overcomes this disadvantage and achieves impressive results. The hybrid factorization structure avoids the calculation hierarchical factorization function. Although the IQL value of $Q_{\mathcal{G}_m}$ following (11) does not equal the factorized value of $Q_{\mathcal{G}_m}$ following (9), the monotonicity of factorization and GIGM still hold. As a result, the optimal policy of GHQ converge to the same optimal policy provided by the fully factorized structure. In conclusion, the total loss of GHQ is written below, an overall framework of GHQ algorithm is illustrated in Fig.1, and the full procedure of GHQ is shown in pseudo-code in Appendix 3.

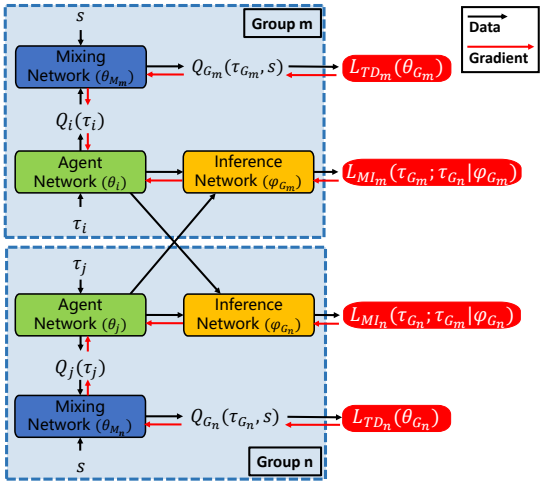

Figure 1: An overall framework of GHQ. $\boldsymbol{\theta}_{\mathcal{G}_m}$ of group m consists of three parts: agent network θ_i, mixing network θ_{M_m} and inference network $\boldsymbol{\psi}_{\mathcal{G}_m}$. θ_i includes $l_{\mathcal{G}_m}$ and $h_{\mathcal{G}_m}$ for calculating $q_{\psi_{\mathcal{G}_m}}$, and also generates Q_i for choosing actions. θ_{M_m} takes Q_i and s for calculating TD loss $\mathcal{L}_{TD_m}$ with hybrid factorization. $\boldsymbol{\psi}_{\mathcal{G}_m}$ takes $l_{\mathcal{G}_m}$, $h_{\mathcal{G}_m}$ and $l_{\mathcal{G}_n}$ for calculating IGMI loss $\mathcal{L}_{MI_m}$.

$$\mathcal{L}_{GHQ}(\boldsymbol{\theta},\boldsymbol{\psi})=\lambda_{TD}\mathcal{L}_{TD}(\boldsymbol{\theta})+\lambda_{MI}\mathcal{L}_{MI}(\boldsymbol{\theta},\boldsymbol{\psi})\,.\tag{13}$$

6 Experiments and Results

6.1 Designing Asymmetric Heterogeneous Maps

Symmetric heterogeneous maps are very common in original SMAC maps (Appendix 1). But it is "unfair" for the internal AI script of StarcraftII, because it is incapable of coordinating and collaborating among multiple agent types. The basic acting pattern of the internal script is "to attack all enemies in sight". It cannot leverage the differences of *shot-range*, position and other properties of agents to divide total damage equally among all units, reduce casualties and produce more damage. The result is that symmetric heterogeneous maps are not as hard as they are expected to be.

Our new asymmetric heterogeneous maps avoid the shortage of original maps. For the agent side, we have Marine and Medivac, a U_{atk} on the ground and a U_{spt} in the air. For the enemy side, we have only Marine to prevent the incapability of the script and increase the amount of Marine to balance the difficulty. Lots of pre-experiments are implemented to determine the specific number of all units. Table 1 shows the information of all new maps.

6.2 Influence of LTH on the Exchange Ratio

According to our analysis in section 4, the existence of LTH is clear, but analysing its influence on agent policy is still

Map Name	Ally_ Marines	Ally_ Medivacs	Enemy_ Marines	Difficulty
6m2m_15m	6	2	15	Easy
6m2m_16m	6	2	16	Medium
8m3m_21m	8	3	21	Medium
8m4m_23m	8	4	23	Hard
12m4m_30m	12	4	30	Ex-Hard
15m2m_28m	15	2	28	Hard
16m2m_30m	16	2	30	Ex-Hard

Table 1: New Asymmetric Heterogeneous SMAC Maps.

Map Name	Ally_ Marine	Ally_ Medivac	Enemy_ Marine	*ER*	*WR*
11m_15m	11	0	15	1:1.36	0.0
12m_15m	12	0	15	**1:1.25**	**0.5**
13m_15m	13	0	15	1:1.15	1.0
15m_20m	15	0	20	1:1.33	0.0
16m_20m	16	0	20	**1:1.25**	**0.5**
17m_20m	17	0	20	**1:1.18**	**1.0**
24m_30m	24	0	30	**1:1.25**	**0.5**
25m_30m	25	0	30	1:1.20	0.9
26m_30m	26	0	30	1:1.15	1.0
6m2m_15m	6	2	15	1:2.50	0.8
8m3m_21m	8	3	21	1:2.63	0.8
15m2m_28m	15	2	28	1:1.87	0.8
17m2m_30m	17	2	30	1:1.76	0.9
7m2m_15m	7	2	15	**1:2.14**	**1.0**
8m3m_19m	8	3	19	**1:2.38**	**1.0**
16m2m_28m	16	2	28	**1:1.75**	**1.0**

Table 2: Exchange Ratio and Winning Rate on Homogeneous and Heterogeneous maps.

required and therefore we need an objective index. The *Winning Rate (WR)* is the probability of MARL agents eliminating all enemies and winning the game, and is approximated by the frequency of winning. The *Exchange Ratio (ER)* is the ratio of the number of weighted attacking units of two sides:

$$ER = \frac{\sum_{all\ types} w_i \cdot |U_{A_i}|}{\sum_{all\ types} w_e \cdot |U_{A_e}|} = \frac{ally\ Marines}{enemy\ Marines}, \quad (14)$$

where $|U_{A_i}|$ and $|U_{A_e}|$ are the number of different types of attacking units U_{atk} for agent side and enemy side, and w_i and w_e are correction weights. In our maps, since the only U_{atk} is Marine, *ER* equals to the ratio of the number of Marines of two sides. We consider that *ER* is an objective index to quantify and compare the difficulty of different maps with similar unit scale and property, and thus is proper to show the influence of LTH.

We design additional homogeneous maps consist of only Marine unit for both sides. The enemy side consists of 15, 20, 30 Marines respectively and remains unchanged, which is almost the same as our heterogeneous maps' settings. The agent side consists of Marines slightly less than the enemy side, as is shown in Table 2. According to the converged *WR*, we conclude that in homogeneous maps with only Marine unit and QMIX agents, *ER* and *WR* are highly related and proportional. When *ER* is about $1:1.25$, *WR* is about 0.5; and when *ER* is about $1:1.18$, *WR* is about 1.0. Even though the total number of units is doubled, this relation remains unchanged. If *ER* keeps increasing to more than about $1:1.3$, *WR* tends to become zero. If *ER* decreases to about $1:1$, as in symmetric maps, it is relatively easy for any of the MARL policy to win. This proves the "unfair" of original SMAC symmetric maps.

On the other hand, both GHQ and QMIX are capable of increasing *ER* up to about $1:1.7$ to $1:2.4$ when *WR* is about 1.0 for asymmetric heterogeneous maps, which suggests that the influence of LTH is objective and should not be discarded. Following experiments show that better utilizing LTH helps GHQ to acquire higher *WR* with smaller variance. Additionally, we further conclude that the "strength" of 1 Medivac equals to about 3 to 4 Marines.

6.3 Asymmetric Heterogeneous MARL Baseline

Experiments are taken in our seven new maps (Table 1) and the MMM2 map as an original asymmetric heterogeneous map. Due to the discrete property of SMAC, value-based methods have achieved better results than policy-based methods [Rashid *et al.*, 2018; Hu *et al.*, 2021; Yu *et al.*, 2021a]. Therefore, we mainly choose value-based methods to run experiments, including vanilla QMIX [Rashid *et al.*, 2018], fine-tuned QMIX (QMIX-FT) [Hu *et al.*, 2021], QPLEX [Wang *et al.*, 2020a], ROMA [Wang *et al.*, 2020b], RODE [Wang *et al.*, 2020c], MAIC [Yuan *et al.*, 2022] and CDS [Li *et al.*, 2021]. Results for extra policy-based algorithms are shown in Appendix 7.

For all algorithms, we use the official implementation with minimal necessary adaptation to our new environmental settings. We use the averaged *WR* of 32 testing episodes as our evaluation index. Testing episodes are taken every 10,000 training steps (about 1,000 training episodes). 5 rounds of complete experiments with different random seeds are performed for plotting the curve of the averaged *WR* with the p-value being 0.95. In general, ROMA cannot learn effective policy within 5M training-steps, because the default training step of ROMA is 20M. RODE is very sensitive to the hyperparameter *n_role_clusters*, and we set it to 5 for all tests according to the original article. MAIC and CDS suffer from LTH and cannot acquire high *WR* with small variance.

All of our results are in red color and other value-based algorithms are shown in the legend. The full experiments can be divided into four sub-groups: (1) validating test on original map, (2) influence of increasing Medivac, (3) symmetric scaling, and (4) asymmetric scaling. The results of every subgroup are shown in Fig. 2.

Validating Test on Original Map

The results of all algorithms on MMM2 are shown in Fig.2 (a). Because the map is relatively easy and almost all algorithms are converged at 3M training steps, we only show the results ended at 3M steps for better analysing. The graph shows that *WR* of most algorithms converged to 1.0 at about 1.5M steps with relatively small variance. QPLEX and GHQ

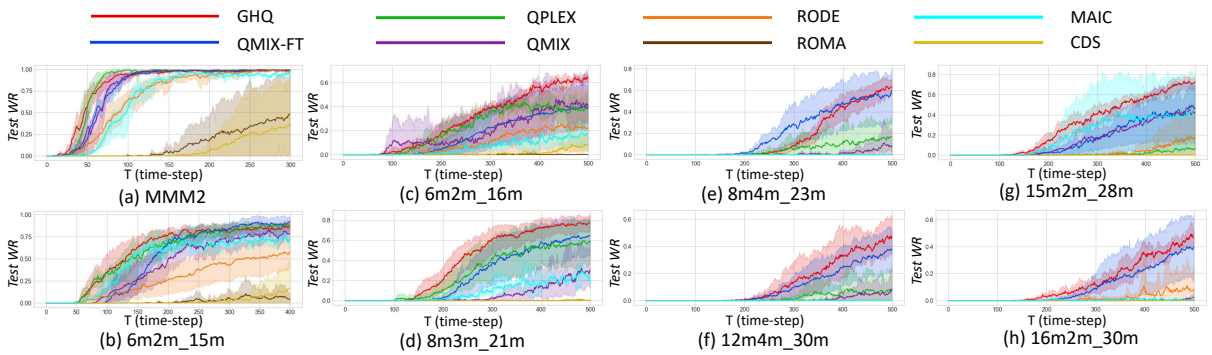

Figure 2: Results for Asymmetric Heterogeneous MARL Baseline.

are slightly better than QMIX. RODE and MAIC converge at about 2.5M steps which is slower than other methods, while ROMA and CDS fail to converge at 3M steps.

Influence of Increasing Medivac

According to section 6.2, the influence of LTH is clear. Furthermore, it can be concluded that increasing the proportion of Medivac in the MAS leads to the increase of policy divergence, and methods using parameter sharing among all agents should be effected. The results are shown in Fig.2 (c), (d), and (e). GHQ achieves best results in all maps with relatively small variance, indicating the effectiveness of the LTG method and IGMI Loss. QMIX-FT achieves similar *WR* against GHQ but has suffered from high variance. QPLEX performs well in (c) and (d), but it's *WR* decreases evidently in (e), indicating the influence of LTH. *WR* of RODE is about 0.2 in (c), but remains zero in other maps. ROMA fails to learn effective policy in all maps. MAIC has a *WR* of about 0.2 in (c) and (d), but fails in (e). CDS has little *WR* in (c) but fails in other maps. The results further prove the necessity to study and utilize LTH.

Symmetric Scaling

Symmetric scaling means to scale up all units in the map simultaneously. The main challenge is to deal with the scalability, because theoretically, the optimal policies of two maps are similar and learnable. The results are shown in Fig.2 (b) and (f). In (b), most algorithms achieve high *WR* within 5M steps, while GHQ converges fastest and RODE suffers from high variance and relatively low *WR*. ROMA and CDS fail to learn effective policy in (b). However, in (f), almost all algorithms fail to learn effective policy, indicating their suffering from scalability problem. GHQ and QMIX-FT perform best and have not yet converged at 5M steps. It is possible that they may converge to the similar policy as their policy in small scale map (b) after enough training steps. QPLEX also suffers from the scalability problem, but generally performs better than QMIX, ROMA, RODE, MAIC and CDS.

Asymmetric Scaling

Asymmetric scaling means to increase Marine of both sides, while maintaining the number of Medivac being 2. As analyzed before, increasing Marine equals to decreasing the proportion of Medivac in the MAS, which leads to the decrease of policy divergence, and methods using parameter sharing among all agents may learn better policy than the setting of increasing Medivac. The results are shown in Fig.2 (b), (c), (g), and (h). Even though algorithms perform well in small scale maps (b) and (c), only GHQ and QMIX-FT perform well in both of the large scale maps (g) and (h), and GHQ outperforms QMIX-FT with smaller variance. MAIC and QMIX perform well in (g) but fail in (h), indicating that the influence of LTH requires MI between different entities to improve cooperation. QPLEX and RODE cannot learn effective policy in (g) and (h), while ROMA and CDS fails in (g) and (h). RODE performs better than symmetric scaling (f), indicating the training of role-selector requires homogeneous MAS settings. Furthermore, it can be concluded that in heterogeneous settings, algorithms in small scale maps may achieve higher *ER*, and asymmetric scaling would decrease *ER*. If we want to maintain the high *ER*, we need to increase Medivac simultaneously and therefore the diversity of agent policies and the influence of LTH remain and even increase.

6.4 Ablation Study

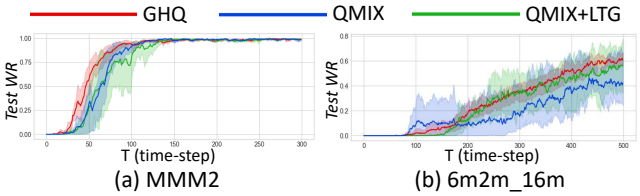

Figure 3: Results for ablation experiments.

Ablation study is taken in MMM2 (a) and 6m2m_16m (b), showing the effectiveness of LTG method and IGMI loss. We choose vanilla QMIX and QMIX+LTG as the ablation groups. The results are shown in Fig.3. The curves show that adding LTG or IGMI loss helps to improve the performance of QMIX, and adding both of them to form the GHQ method further improve *WR*.

6.5 Visualization of the Terminal Policy

The terminal policy is the policy determined by the network parameters at the end of the whole training process, which is 5M training steps in our experiments. We analyse the difference in choosing action and *health-points* of the two

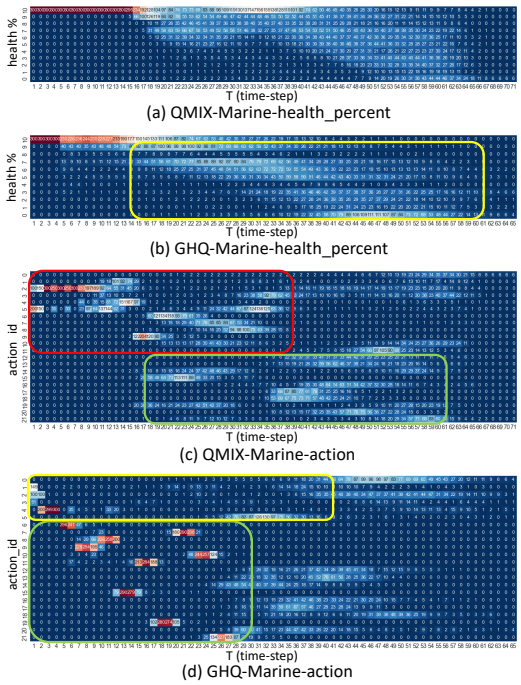

Figure 4: Marines' heat-map for the terminal policy in 6m2m_16m.

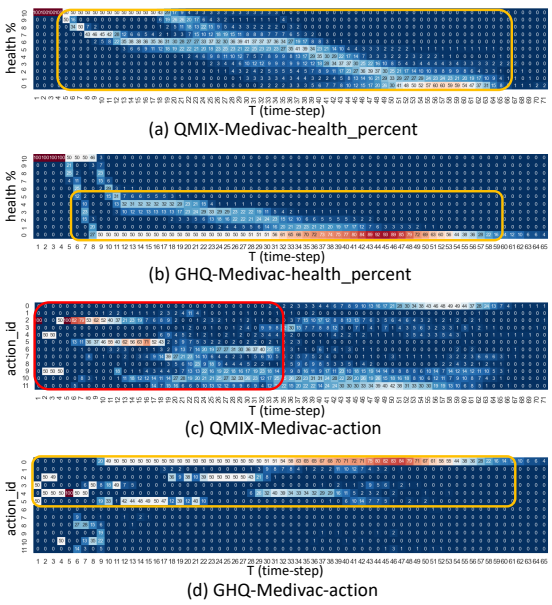

Figure 5: Medivacs' heat-map for the terminal policy in 6m2m_16m.

unit types controlled by GHQ and QMIX-FT with the same hyper-parameter and similar network capacity. We choose the terminal policies in 6m2m_16m, test them for 50 times and record their trajectories. We calculate the sum of agents' chosen-actions and agents' percentage of *health-points* of different groups and visualize in two heat-maps about Marines (Fig.4) and Medivacs (Fig.5) separately. The horizontal coordinate of heat-maps is the time-step T, the vertical coordinate is the action ID number or the every 10^{th} percentile of *health-points*, and the temperature is the sum of corresponding agents. In action heat-maps, action ID 0 to 5 are *common actions* A_{com} for moving and stopping, while the rest are *interactive actions* A_{act} for attacking or healing. Chosen parameters of GHQ and QMIX-FT reach the same *WR* of about 0.8 after 5M training steps, noting that GHQ learns faster than QMIX-FT. Figures clearly show that the two algorithms achieve similar result through different agent policy. We conclude three key findings:

- *Parameter sharing* among different agent types do influence agent policy. As is suggested in [Kuba *et al.*, 2021], parameter sharing restricts network parameters from being diverse. Red boxes in Fig.4 (c) and Fig.5 (c) show similar policy pattern of "first move and then stop to attack/heal" for the two agent types in QMIX-FT. Agents also prefer to choose action 2 and 5 in the first 15 time-steps. In GHQ, however, LTG method guarantees the diversity of different network parameters from different groups, as is shown in Fig.4 (d) and Fig.5 (d).
- GHQ improves group policy learning. Green boxes in Fig.4 (c) and (d) indicate that Marine controlled by GHQ learns better "focus-firing" policy, as the temperature of A_{act} are notably hotter than QMIX-FT, in which agents learn to fire at several targets at the same time. Yellow box in Fig.4 (d) shows that moving policies of GHQ Marines are significantly different from QMIX-FT. GHQ Marines finish their movement in the first 4 time-steps with decisive actions and form a tight front. They tend to stay together and therefore take enemy damage simultaneously, which leads to similar decreasing tendency of *health-point* and the two obvious temperature valleys at 80 and 40 percentile in yellow box of Fig.4 (b).
- GHQ improves inter-group cooperating. Orange boxes in Fig.5 (a) and (b) represent the decreasing curves of Medivacs' *health-point*. GHQ Medivacs have learned better "distracting" policy than QMIX-FT Medivacs. One GHQ Medivac firstly moves towards enemies and attracts fire to prevent enemies attacking allies, as is proved in orange box in (d) with the "action 0 line" indicating the death of Medivac. After its death, the other Medivac moves on to keep attracting enemy fire, and therefore the curve in (b) consists of two independent curves. The distraction policy performed by GHQ Medivacs is a fabulous tactic and differes from GHQ Marines' policy, indicating that GHQ is capable of utilizing LTH for better cooperation.

7 Conclusion

In this paper, we give formal definition of LTH and study LTH in SMAC with newly designed maps. We propose GIGM and GHQ algorithm to solve LTH preliminarily. Experiments show that GHQ outperforms other state-of-the-art algorithms. We believe that the study of heterogeneity is indispensable for complex MARL. In future, we will try to solve large scale and complex heterogeneous MARL problems in other maps and environments.

References

[Al Faiya *et al.*, 2021] Badr Al Faiya, Dimitrios Athanasiadis, Minjiang Chen, Stephen McArthur, Ivana Kockar, Haowei Lu, and Francisco De Leon. A self-organizing multi-agent system for distributed voltage regulation. *IEEE Transactions on Smart Grid*, 12(5):4102–4112, 2021.

[Bono *et al.*, 2018] Guillaume Bono, Jilles Steeve Dibangoye, Laëtitia Matignon, Florian Pereyron, and Olivier Simonin. Cooperative multi-agent policy gradient. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 459–476, 2018.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*, 2014.

[Dong *et al.*, 2021] Heng Dong, Tonghan Wang, Jiayuan Liu, Chi Han, and Chongjie Zhang. Birds of a feather flock together: A close look at cooperation emergence via multi-agent rl. *arXiv preprint arXiv:2104.11455*, 2021.

[Foerster *et al.*, 2016] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[Foerster *et al.*, 2017] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. *International conference on machine learning*, pages 1146–1155, 2017.

[Gronauer and Diepold, 2022] Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022.

[Guestrin *et al.*, 2001] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored mdps. *Advances in neural information processing systems*, 14, 2001.

[Gupta *et al.*, 2017] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. *International conference on autonomous agents and multiagent systems*, pages 66–83, 2017.

[Hartmann *et al.*, 2021] Valentin Noah Hartmann, Andreas Orthey, Danny Driess, Ozgur S Oguz, and Marc Toussaint. Long-horizon multi-robot rearrangement planning for construction assembly. *arXiv preprint arXiv:2106.02489*, 2021.

[Hernandez-Leal *et al.*, 2019] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797, 2019.

[Hou *et al.*, 2022] Jialiang Hou, Xin Zhou, Zhongxue Gan, and Fei Gao. Enhanced decentralized autonomous aerial swarm with group planning. *arXiv preprint arXiv:2203.01069*, 2022.

[Hu *et al.*, 2021] Jian Hu, Siyang Jiang, Seth Austin Harding, Haibin Wu, and Shih-wei Liao. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv e-prints*, pages arXiv–2102, 2021.

[Ivić, 2020] Stefan Ivić. Motion control for autonomous heterogeneous multiagent area search in uncertain conditions. *IEEE Transactions on Cybernetics*, 2020.

[Kraemer and Banerjee, 2016] Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

[Kuba *et al.*, 2021] Jakub Grudzien Kuba, Ruiqing Chen, Munning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

[Li *et al.*, 2021] 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.

[Li *et al.*, 2022] Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E Taylor, and Zhen Wang. Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration. *arXiv preprint arXiv:2203.08553*, 2022.

[Ling *et al.*, 2022] Zhiwei Ling, Zhihao Yue, Jun Xia, Ming Hu, Ting Wang, and Mingsong Chen. Fedentropy: Efficient device grouping for federated learning using maximum entropy judgment. *arXiv preprint arXiv:2205.12038*, 2022.

[Mahajan *et al.*, 2019] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.

[Oliehoek and Amato, 2016] Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. *International conference on machine learning*, 2018.

[Rashid *et al.*, 2020a] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10199–10210, 2020.

[Rashid *et al.*, 2020b] 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. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

[Rotman *et al.*, 2020] Daniel Rotman, Yevgeny Yaroker, Elad Amrani, Udi Barzelay, and Rami Ben-Ari. Learnable optimal sequential grouping for video scene detection. *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1958–1966, 2020.

[Samvelyan *et al.*, 2019] 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.

[Son *et al.*, 2019] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. *International conference on machine learning*, pages 5887–5896, 2019.

[Sunehag *et al.*, 2017] 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.

[Tan, 1993] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

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

[Wang *et al.*, 2020b] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020.

[Wang *et al.*, 2020c] 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.

[Yang and Parasuraman, 2021] Qin Yang and Ramviyas Parasuraman. How can robots trust each other for better cooperation? a relative needs entropy based robot-robot trust assessment model. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2656–2663, 2021.

[Yang *et al.*, 2020] 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.

[Yoon *et al.*, 2019] Hyung-Jin Yoon, Huaiyu Chen, Kehan Long, Heling Zhang, Aditya Gahlawat, Donghwan Lee, and Naira Hovakimyan. Learning to communicate: A machine learning framework for heterogeneous multi-agent robotic systems. *AIAA Scitech 2019 Forum*, page 1456, 2019.

[Yu *et al.*, 2021a] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

[Yu *et al.*, 2021b] Yiding Yu, Soung Chang Liew, and Taotao Wang. Multi-agent deep reinforcement learning multiple access for heterogeneous wireless networks with imperfect channels. *IEEE Transactions on Mobile Computing*, 2021.

[Yuan *et al.*, 2022] Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. Multi-agent incentive communication via decentralized teammate modeling. *Association for the Advancement of Artificial Intelligence*, 2022.