Greedy based Value Representation for Optimal Coordination in Multi-agent Reinforcement Learning

Lipeng Wan [* 1] **Zeyang Liu** [* 1] **Xingyu Chen** [1] **Xuguang Lan** [1] **Nanning Zheng** [1]

Abstract

Due to the representation limitation of the joint Q value function, multi-agent reinforcement learning methods with linear value decomposition (LVD) or monotonic value decomposition (MVD) suffer from relative overgeneralization. As a result, they can not ensure optimal consistency (i.e., the correspondence between individual greedy actions and the maximal true Q value). In this paper, we derive the expression of the joint Q value function of LVD and MVD. According to the expression, we draw a transition diagram, where each self-transition node (STN) is a possible convergence. To ensure optimal consistency, the optimal node is required to be the unique STN. Therefore, we propose the greedy-based value representation (GVR), which turns the optimal node into an STN via inferior target shaping and further eliminates the non-optimal STNs via superior experience replay. In addition, GVR achieves an adaptive trade-off between optimality and stability. Our method outperforms state-of-the-art baselines in experiments on various benchmarks. Theoretical proofs and empirical results on matrix games demonstrate that GVR ensures optimal consistency under sufficient exploration.

1. Introduction

By taking advantage of the deep learning technique, cooperative multi-agent reinforcement learning (MARL) shows great scalability and excellent performance on challenging tasks (Vorotnikov et al., 2018; Wu et al., 2020) such as Star-Craft unit micromanagement (Foerster et al., 2018). An essential problem of cooperative MARL is credit assignment. As a popular approach to address the problem, value

decomposition gains growing attention. The main concern of value decomposition is the optimality of coordination. In a successful case of credit assignment via value decomposition, agents perform individual greedy actions according to their local utility functions and achieve the best team performance (i.e., the optimal true Q value). Here we define the correspondence between the individual greedy actions and the optimal true Q value as *optimal consistency*.

Due to the representation limitation of the joint Q value function, linear value decomposition (LVD) or monotonic value decomposition (MVD) suffer from relative overgeneralization (RO) (Panait et al., 2006; Wei et al., 2018). As a result, they can not ensure the optimal consistency. Recent works address RO from two different perspectives. The first kind of method completes the representation capacity of the joint Q value function (e.g., QTRAN (Son et al., 2019) and QPLEX (Wang et al., 2020)). However, learning the complete representation is impractical in complicated MARL tasks because the joint action space increases exponentially with the number of agents. The other kind of method tries to prevent sub-optimal convergences by learning a biased joint Q value function (e.g., WQMIX (Rashid et al., 2020) and MAVEN (Mahajan et al., 2019)), which depends on heuristic parameters and is only applicable in specific tasks. More discussions about RO and related works are provided in Appendix A.

In this paper, we derive the expression of the joint Q value function of LVD and MVD and draw some interesting conclusions. Firstly, LVD and MVD share the same expression of the joint Q value function. Secondly, the joint Q value of any action depends on the true Q values of all actions in the whole joint action space. Thirdly, the joint Q values transfer with greedy actions, by which a transition diagram is acquired. In the transition diagram, each self-transition node (STN) is a possible convergence. A node that satisfies the optimal consistency is called an optimal node, otherwise, we call it a non-optimal node. To ensure the optimal consistency, the optimal node is required to be the unique STN, which is the *target problem* of this paper.

To address the target problem, we propose the greedy-based value representation (GVR). Firstly, we reshape the representation target of the inferior actions (i.e., the actions with

---

*Equal contribution [1]School of Artificial Intelligence, Xian Jiaotong University, Xian, Shaanxi, China. Correspondence to: Xuguang Lan <xglan@mail.xjtu.edu.cn>.

poorer performance than current greedy), which is prove to ensure that the optimal node would always be an STN. We also prove that under the inferior target shaping, non-optimal nodes would be eliminated when the probabilities of superior actions (i.e., the actions with poorer performance than current greedy) exceed a threshold. Therefore, we introduce superior experience replay, which steadily raises the proportions of superior actions in the training batch. It is proved that GVR ensures the optimal consistency under sufficient exploration. However, excessive pursuit for optimality would weaken the stability in tasks with multiple optimums or multiple approximative optimums, for which we further design an adaptive trade-off between optimality and stability in GVR.

We have three contributions in this work. (1) This is the first work to derive the general expression of the joint Q value function for LVD and MVD. (2) Based on the expression of the joint Q value function, we draw a transition diagram and propose a quantified condition to ensure the optimal consistency for LVD and MVD. (3) We propose the GVR algorithm. GVR ensures the optimal consistency under sufficient exploration. Besides, GVR achieves an adaptive trade-off between optimality and stability. Our method outperforms state-of-the-art baselines in various benchmarks.

2. Preliminaries

2.1. Dec-POMDP

We model a fully cooperative multi-agent task as a decentralized partially observable Markov decision process (Dec-POMDP) described by a tuple $\mathcal{G} =< S, U, P, r, Z, O, n, \gamma >$ (Guestrin et al., 2001; Oliehoek & Amato, 2016). $s \in S$ denotes the true state of the environment. At each time step, each agent $a \in A \equiv \{1, 2, \cdots, n\}$ receives a local observation $z^a \in Z$ produced by the observation function $O : S \times A \rightarrow Z$, and then chooses an individual action $u^a \in U$ according to a local policy $\pi^a(u^a|\tau^a) : T \times U \rightarrow [0, 1]$, where $\tau^a \in T \equiv (Z \times U)^*$ denotes the local action-observation history. The joint action of n agents $\mathbf{u}$ results in a shared reward $r(s, \mathbf{u})$ and a transition to the next state $s' \sim P(\cdot|s, \mathbf{u})$. $\gamma \in [0, 1)$ is a discount factor.

We denote the joint variable of group agents with bold symbols, e.g., the joint action $\boldsymbol{u} \in \boldsymbol{U} \equiv U^n$, the joint action-observation history $\boldsymbol{\tau} \in \boldsymbol{\mathcal{T}} \equiv T^n$, and the joint policy (i.e., the policy interacts with environment to generate trajectories) $\boldsymbol{\pi}(\mathbf{u}|\boldsymbol{\tau})$. The *true Q value* is denoted by $\mathcal{Q}(s_t, \boldsymbol{u}_t) = \mathbb{E}_{s_{t+1:\infty}, \boldsymbol{u}_{t+1:\infty}} [R_t|s_t, \boldsymbol{u}_t]$, where $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+1}$ is the discounted return. The action-state value function of agent a and the group of agents are defined as *utility function* $\mathcal{U}^a(u^a, \tau^a)$ and *joint Q value function* $Q(\boldsymbol{u}, \boldsymbol{\tau})$ respectively.

The true Q value is the target of the joint Q value in training, which is the unique external criterion of the team performance. The *greedy action* $\hat{\boldsymbol{u}} := argmax_{\boldsymbol{u}} Q(\boldsymbol{u}, \boldsymbol{\tau})$ is defined as the joint action with the maximal joint Q value . The *optimal action* $\boldsymbol{u}^* := argmax_{\boldsymbol{u}} \mathcal{Q}(s, \boldsymbol{u})$ is defined as the joint action with the best team performance. For brevity, we sometimes omit the prefix "joint" for the joint variables.

2.2. Optimal Consistency and TGM Principle

In centralized training with decentralized execution (CTDE) (Oliehoek et al., 2008; Foerster et al., 2016; Lowe et al., 2017), agents are expected to act individually according to their local policies (i.e., the individual greedy actions) while achieving the best team performance (i.e., the optimal true Q value). Here we define the correspondence between the individual greedy actions and the optimal true Q value as the optimal consistency.

Definition 1 (Optimal consistency). Given a set of utility functions $\{\mathcal{U}^1(u^1, \tau^1)), \cdots, \mathcal{U}^n(u^n, \tau^n)\}$, and the true Q value $\mathcal{Q}(s, \boldsymbol{u})$, if the following holds

$$\{\underset{u^1}{argmax}\, \mathcal{U}^1(u^1, \tau^1), \cdots, \underset{u^n}{argmax}\, \mathcal{U}^n(u^n, \tau^n)\}$$
$$= \underset{\boldsymbol{u}}{argmax}\, \mathcal{Q}(s, \boldsymbol{u}) \tag{1}$$

then we say the set of utility functions $\{\mathcal{U}^1(u^1, \tau^1)), \cdots, \mathcal{U}^n(u^n, \tau^n)\}$ satisfies the optimal consistency. For simplicity, we ignore situations with non-unique optimal actions.

The optimal consistency can be decomposed into two principles: Individual-Global-Max (IGM) and True-Global-Max (TGM). The IGM principle proposed by QTRAN (Son et al., 2019) is defined on the correspondence between individual greedy actions and the joint greedy actions (formally, $\{argmax_{u^1}\, \mathcal{U}^1(u^1, \tau^1), \cdots, argmax_{u^n}\, \mathcal{U}^n(o^n, \tau^n)\} = argmax_{\boldsymbol{u}} Q(\boldsymbol{u}, \boldsymbol{\tau})$). To achieve the optimal consistency, the correspondence between the joint greedy action and the optimal true Q value is required, for which we define the TGM principle:

Definition 2 (TGM). Given a joint value function $Q(\boldsymbol{u}, \boldsymbol{\tau})$, and the true Q value $\mathcal{Q}(s, \boldsymbol{u})$, if the following holds

$$\underset{\boldsymbol{u}}{argmax}\, Q(\boldsymbol{u}, \boldsymbol{\tau}) = \underset{\boldsymbol{u}}{argmax}\, \mathcal{Q}(s, \boldsymbol{u}) \tag{2}$$

then we say the joint value function $Q(\boldsymbol{u}, \boldsymbol{\tau})$ satisfies the TGM principle. For simplicity, we ignore situations with non-unique optimal actions.

3. Investigation of the TGM Principle for LVD & MVD

Linear value decomposition (LVD) and monotonic value decomposition (MVD) naturally meet the IGM principle
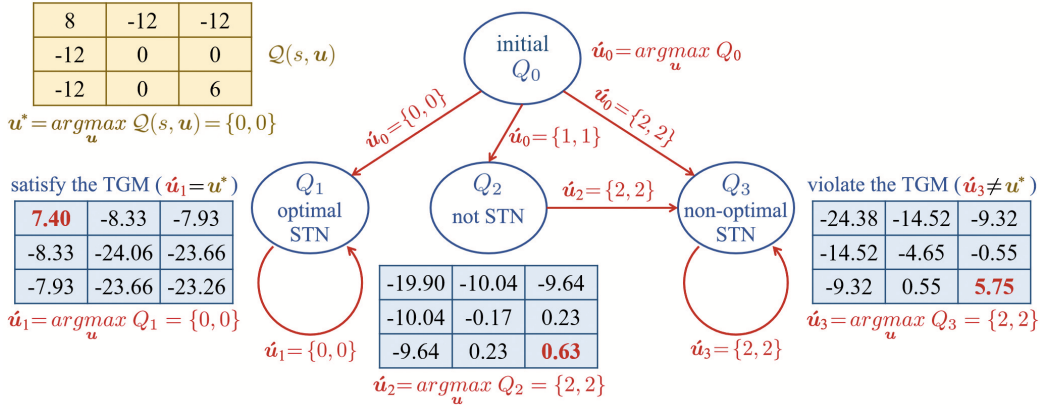
Figure 1. Transition diagram of $Q(\boldsymbol{u}, \boldsymbol{\tau})$ (blue tables, calculated by Eq.4). The true Q values $\mathcal{Q}(s, \boldsymbol{u})$ are shown in the yellow table. Initial $Q(\boldsymbol{u}, \boldsymbol{\tau}) = Q_0$ transfers to different nodes (i.e., Q_1, Q_2, Q_3) according to the corresponding greedy action $\acute{\boldsymbol{u}}_0 = argmax_{\boldsymbol{u}} Q_0$. Q_2 is not a STN since it finally transfers to Q_3. Q_1 is the optimal STN while Q_3 is a non-optimal STN. We omit the other cases of $\acute{\boldsymbol{u}}_0$ such as $\acute{\boldsymbol{u}}_0 = \{0, 1\}$ and $\acute{\boldsymbol{u}}_0 = \{2, 0\}$.

(Son et al., 2019). To achieve the optimal consistency, we investigate the conditions of the TGM principle for LVD and MVD. In this section, we first derive the expression of $Q(\boldsymbol{u}, \boldsymbol{\tau})$ (Eq.4), by which we draw a transition diagram (Fig.1) of the joint Q values. In this diagram, each self-transition node (STN) is a possible convergence, where the TGM principle is satisfied by the optimal node but violated by the non-optimal nodes. Finally, we propose a sufficient condition (Eq.6) of the TGM principle, which ensures the optimal node is the unique STN.

3.1. Expression of the Joint Q Value Function for LVD & MVD

Firstly, take two-agent LVD as an example, where the joint Q value function $Q(u_i^1, u_j^2, \boldsymbol{\tau})$ is linearly factorized into two utility functions as $Q(u_i^1, u_j^2, \boldsymbol{\tau}) = \mathcal{U}^1(u_i^1, \tau^1) + \mathcal{U}^2(u_j^2, \tau^2)$. $u_i^1, u_j^2 \in \{u_1, \cdots, u_m\}$ are the actions of agents 1 and 2, respectively, where $\{u_1, \cdots, u_m\}$ is the discrete individual action space. The greedy actions of two agents are denoted with $\acute{u}^1$ and $\acute{u}^2$. For briefness, we denote $\mathcal{Q}(s, u_i^1, u_j^2)$ and $\mathcal{U}^a(u_i^a, \tau^a)$ with $\mathcal{Q}_{ij}$ and $\mathcal{U}_i^a(a \in \{1, 2\})$, respectively. Since the utility functions are trained to approximate different true Q values in different combinations, under $\epsilon-$greedy visitation, we have

$$\mathcal{U}_i^1 = \frac{\epsilon}{m} \sum_{k=1}^{m}(\mathcal{Q}_{ik} - \mathcal{U}_k^2) + (1 - \epsilon)(\mathcal{Q}_{i\acute{j}} - \mathcal{U}_{\acute{j}}^2)$$

$$\mathcal{U}_j^2 = \frac{\epsilon}{m} \sum_{k=1}^{m}(\mathcal{Q}_{kj} - \mathcal{U}_k^1) + (1 - \epsilon)(\mathcal{Q}_{\acute{i}j} - \mathcal{U}_{\acute{i}}^1) \tag{3}$$

where the subscript $\acute{i}$ and $\acute{j}$ refer to the greedy actions of two agents. Through the derivation provided in Appendix

B, the expression of $Q(u_i^1, u_j^2, \boldsymbol{\tau})$ can be acquired

$$Q(u_i^1, u_j^2, \boldsymbol{\tau}) = \frac{\epsilon}{m} \sum_{k=1}^{m}(\mathcal{Q}_{ik} + \mathcal{Q}_{kj}) + (1 - \epsilon)(\mathcal{Q}_{i\acute{j}} + \mathcal{Q}_{\acute{i}j})$$
$$- \frac{\epsilon(1 - \epsilon)}{m} \sum_{k=1}^{m}(\mathcal{Q}_{\acute{i}k} + \mathcal{Q}_{k\acute{j}})$$
$$- \frac{\epsilon^2}{m^2} \sum_{i=1}^{m}\sum_{j=1}^{m}\mathcal{Q}_{ij} - (1 - \epsilon)^2 \mathcal{Q}_{\acute{i}\acute{j}} \tag{4}$$

Notice the term $\sum_{i=1}^{m}\sum_{j=1}^{m}\mathcal{Q}_{ij}$ depends on the true Q values of all actions in the whole joint action space. For MVD, the expression of $Q(u_i^1, u_j^2, \boldsymbol{\tau})$ is identical to Eq.4 (the proof is provided in Appendix C.1). Verification of Eq.4 is provided in Appendix C.2. For situations with more than two agents, by referring to the derivation in Appendix B and C.1, the expression of joint Q values can also be obtained.
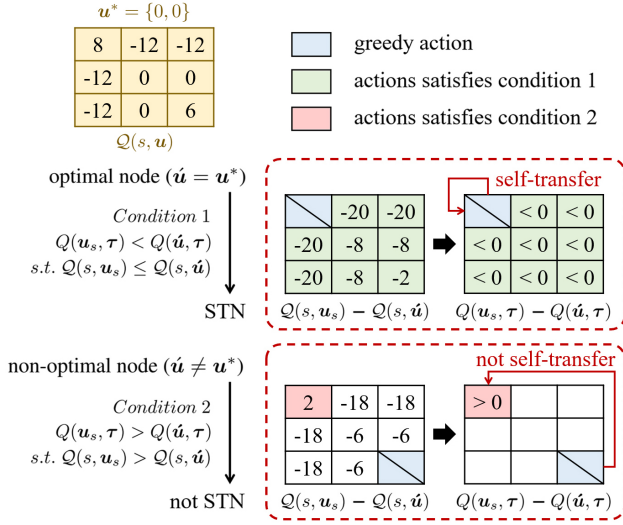
3.2. A Sufficient Condition of the TGM Principle for LVD & MVD

According to Eq.4, the joint Q values $Q(\boldsymbol{u}, \boldsymbol{\tau})(\boldsymbol{u} = \{u_i^1, u_j^2\})$ vary as the greedy action $\acute{\boldsymbol{u}} = \{\acute{u}^1, \acute{u}^2\}$, by which a transition diagram of $Q(\boldsymbol{u}, \boldsymbol{\tau})$ is acquired. An example is shown in Fig.1, where $\epsilon = 0.2$.

Notice Q_1 and Q_3 in Fig.1 are both STNs. An STN satisfies

$$argmax_{\boldsymbol{u}} Q(\boldsymbol{u}, \boldsymbol{\tau}) = argmax_{\boldsymbol{u}} Q_{old}(\boldsymbol{u}, \boldsymbol{\tau}) \tag{5}$$

where $Q_{old}(\boldsymbol{u}, \boldsymbol{\tau})$ and $Q(\boldsymbol{u}, \boldsymbol{\tau})$ are the joint Q value functions of the nodes before and after transition, respectively. Especially, if $Q(\boldsymbol{u}, \boldsymbol{\tau})$ is an STN and satisfies the TGM

Figure 2. Examples of joint Q value functions which satisfy *Condition* 1 (upper dotted box) and *Condition* 2 (lower dotted box), where $\boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$. The true Q values are shown in the yellow table.

principle (i.e., $\acute{\boldsymbol{u}} = \boldsymbol{u}^*$), we say it is the optimal STN. Otherwise, if $Q(\boldsymbol{u}, \boldsymbol{\tau})$ is an STN but violates the TGM principle (i.e., $\acute{\boldsymbol{u}} \neq \boldsymbol{u}^*$), we say it is a non-optimal STN. Each STN is a possible convergence. To ensure the TGM principle, *the optimal node is required to be the unique STN*, i.e., the optimal node must be the only possible convergence. To achieve this, we provide a set of sufficient conditions as follows: for $\forall \boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$,

$$
\begin{aligned}
&\# \ Condition \ 1 \\
&Q(\boldsymbol{u}_s, \boldsymbol{\tau}) < Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \qquad s.t. \ \mathcal{Q}(s, \boldsymbol{u}_s) \leq \mathcal{Q}(s, \acute{\boldsymbol{u}}) \\
&\# \ Condition \ 2 \\
&Q(\boldsymbol{u}_s, \boldsymbol{\tau}) > Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \qquad s.t. \ \mathcal{Q}(s, \boldsymbol{u}_s) > \mathcal{Q}(s, \acute{\boldsymbol{u}})
\end{aligned} \tag{6}
$$

Examples which satisfy *Condition* 1 and *Condition* 2 are shown in Fig.2.

***Condition* 1** ensures the optimal node is an STN. Suppose $\acute{\boldsymbol{u}} = argmax_{\boldsymbol{u}} Q_{old}(\boldsymbol{u}, \boldsymbol{\tau})$ equals to the optimal action, i.e., $\acute{\boldsymbol{u}} = \boldsymbol{u}^*$, we have $\forall \boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$,$\mathcal{Q}(s, \boldsymbol{u}_s) < \mathcal{Q}(s, \acute{\boldsymbol{u}})$. According to *Condition* 1, $\forall \boldsymbol{u}_s \neq \boldsymbol{u}^*$, $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) < Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$. By Eq.5, the optimal node is an STN since $argmax_{\boldsymbol{u}} Q(\boldsymbol{u}, \boldsymbol{\tau}) = \boldsymbol{u}^* = argmax_{\boldsymbol{u}} Q_{old}(\boldsymbol{u}, \boldsymbol{\tau})$.

***Condition* 2** ensures a non-optimal node is not STN. Suppose $\acute{\boldsymbol{u}} = argmax_{\boldsymbol{u}} Q_{old}(\boldsymbol{u}, \boldsymbol{\tau})$ is a non-optimal action, $\exists \boldsymbol{u}_s$ such that $\mathcal{Q}(s, \boldsymbol{u}_s) > \mathcal{Q}(s, \acute{\boldsymbol{u}})$. According to *Condition* 2, $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) > Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$. By Eq.5, the non-optimal node is not an STN since $argmax_{\boldsymbol{u}} Q(\boldsymbol{u}, \boldsymbol{\tau}) \neq argmax_{\boldsymbol{u}} Q_{old}(\boldsymbol{u}, \boldsymbol{\tau})$.

4. Greedy-based Value Representation

It is impractical to evaluate the conditions in Eq.6 for LVD or MVD because both $Q(\boldsymbol{u}_s, \boldsymbol{\tau})$ and $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ depend on the true Q values of all actions in the whole joint action space. In this section, we first introduce inferior target shaping (ITS), where $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ becomes independent to the true Q values of inferior actions. We prove that *Condition* 1 always holds under ITS and *Condition* 2 can be further satisfied by superior experience replay (SER). Besides, as discussed in Appendix H.2, excessive pursuit for optimality decreases the stability. In Section 4.3, we introduce a method to achieve an adaptive trade-off between stability and optimality in GVR. An overview of GVR is given in Appendix I.

4.1. Inferior Target Shaping

According to Eq.4, the joint Q values of the optimal node depend on the true Q values of all actions in the whole joint action space. We can turn the optimal node into an STN by modifying some of these true Q values. Since the exact true Q values of non-optimal actions are uninformative, we reshape them with an ITS target $\mathcal{Q}_{its}(s, \boldsymbol{u})$

$$
\begin{aligned}
\mathcal{Q}_{its}(s, \boldsymbol{u}) &= Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \alpha |Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})| \\
s.t. \ \ \mathcal{Q}(s, \boldsymbol{u}) &\leq \mathcal{Q}(s, \acute{\boldsymbol{u}}) * (1 + e_{Q0}) \ and \ \boldsymbol{u} \neq \acute{\boldsymbol{u}}
\end{aligned} \tag{7}
$$

$\mathcal{Q}(s, \boldsymbol{u})$ and $\mathcal{Q}(s, \acute{\boldsymbol{u}})$ can be acquired by estimation, which is discussed in Section 4.3. An action satisfying the constraints in Eq.7 is called an *inferior action*. Similarly, an action $\boldsymbol{u}$ satisfying $\mathcal{Q}(s, \boldsymbol{u}) > \mathcal{Q}(s, \acute{\boldsymbol{u}}) * (1 + e_{Q0})$ and $\boldsymbol{u} \neq \acute{\boldsymbol{u}}$ is called *superior action*. For greedy or superior actions, $\mathcal{Q}_{its}(s, \boldsymbol{u}) = \mathcal{Q}(s, \boldsymbol{u})$. $\alpha \in (0, \infty)$ is a hyperparameter that defines the gap of the joint Q values' targets between the inferior and greedy actions. $e_{Q0} \in [0, \infty)$ is a hyper-parameter that defines the minimum gap of the true Q values between the superior and greedy actions. An example of the ITS target is provided in Appendix D.1. The ITS target simplifies the representation since there is no need to represent the true Q values of inferior actions. Besides, given the greedy action $\acute{\boldsymbol{u}}$ and another action $\boldsymbol{u}_s(\boldsymbol{u}_s \neq \acute{\boldsymbol{u}})$, assuming $\mathcal{Q}(s, \acute{\boldsymbol{u}}) > 0$, we have

$$
\begin{aligned}
Q(\boldsymbol{u}_s, &\boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \\
&= n(\eta_1 - \eta_2) \left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha) Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \right] \\
&\quad + n \eta_1 e_Q \mathcal{Q}(s, \acute{\boldsymbol{u}})
\end{aligned} \tag{8}
$$

where $e_Q = \frac{\mathcal{Q}_{its}(s, \boldsymbol{u}_s) - \mathcal{Q}(s, \acute{\boldsymbol{u}})}{\mathcal{Q}(s, \acute{\boldsymbol{u}})}$, $\eta_1 = (\frac{\epsilon}{m})^{n-1}$, and $\eta_2 = (1 - \epsilon + \frac{\epsilon}{m})^{n-1}$. We provide two different derivations for Eq.8 in Appendix D, and the calculation result is verified in the experimental part (Fig.5(a)).

Notice $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ is *independent* to the true Q values of inferior actions. It is proved in Appendix E.1

that $Condition$ 1 (Eq.6) always holds under ITS, i.e., the optimal node is *always an STN*. Besides, we also prove that $Condition$ 2 holds when

$$\frac{\eta_1}{\eta_2} > \frac{\alpha}{\alpha + e_{Q0}} = \eta_0 \qquad (9)$$

which indicates the non-optimal STNs can be eliminated by *raising $\frac{\eta_1}{\eta_2}$ under ITS*. A simple way to achieve this is improving exploration. Substituting the expression of η_1 and η_2 into Eq.9, we have

$$\epsilon > \frac{m}{(\frac{e_{Q0}}{\alpha})^{\frac{1}{n-1}} + 1 + m - 1} = \epsilon_0 \qquad (10)$$

where ϵ_0 is the lower bound of ϵ. However, as the number of agents (n) and the size of individual action space (m) increases, ϵ_0 grows close to 1 (an example is provided in Fig.5(b)), which suggests improving exploration is inapplicable in tasks with long episodes.

Another way to raise $\frac{\eta_1}{\eta_2}$ is applying a weight $w(w > 1)$ to the superior actions. It is proved in Appendix F.1 that the non-optimal STNs can be eliminated when $w > \frac{\alpha(\eta_2 - \eta_1)}{e_Q \eta_1} = w_0$, where w_0 is the lower bound of w. However, w_0 grows exponentially as the number of agents increases (as verified in Appendix F.2, $w_0 = 659.50$ when $n = 4$), which introduces instability in $Q(\boldsymbol{u}, \boldsymbol{\tau})$.

4.2. Superior Experience Replay

The difficulty of meeting Eq.9 is that η_0 is a constant while $\frac{\eta_1}{\eta_2}$ decreases exponentially with the number of agents. We consider adding a constant to η_1, i.e., adding a constant to the probabilities of the superior actions. Therefore, we introduce an extra buffer, named *superior buffer*, to store the superior actions. Inspired by prioritized experience replay (PER) (Schaul et al., 2015; Zhang & Sutton, 2017), we assign a priority to each trajectory. The training batch consists of two parts: trajectories randomly sampled from the replay buffer and trajectories sampled from the superior buffer with PER.

We apply a weight w_{ser} to the samples from the superior buffer. The probability of sampling the superior actions from the replay buffer is extremely small. Therefore, the proportion of the superior actions in the training batch is mainly determined by w_{ser}, which is a constant. It is proved in Appendix G that under ITS, SER can eliminate the non-optimal STNs by setting w_{ser} to

$$w_{ser} > \frac{\alpha}{e_{Q0}} (\eta_2 - \eta_1)\eta_s - \eta_1 \eta_s \qquad (11)$$

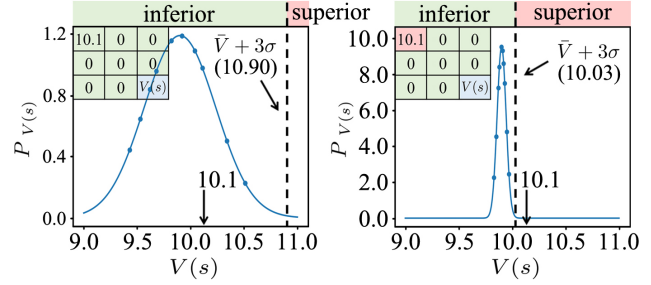where η_s is the probability of state s.

Figure 3. An example of the trade-off between optimality and stability in GVR. According to the outputs of the critics (blue dots), a gaussian distribution (blue curve) is fitted, where the threshold of superior and inferior actions is set to $\bar{V} + 3\sigma$. When $Q(s, \boldsymbol{u}_s) > \bar{V} + 3\sigma$, the probability of u_s being a superior action is 0.9987.

4.3. Adaptive Trade-off between Optimality and Stability

In Eq.7, both $\mathcal{Q}(s, \boldsymbol{u})$ and $\mathcal{Q}(s, \hat{\boldsymbol{u}})$ are unavailable, which require estimation. Excessive pursuit for optimality decreases the stability due to the error of the estimation. An example is provided in Appendix H.2, where the joint Q values frequently transfer among the reckoned "optimal" nodes and poor intermediate nodes. As a result, methods without a trade-off between optimality and stability may suffer from poor performance.

In Eq.7, notice e_{Q0} defines the minimum gap of the true Q values between the superior and greedy actions. When the estimation error of the true Q values exceeds e_{Q0}, an inferior action may be mistaken for a superior action, which causes instability. As e_{Q0} grows, the actions just slightly better than current greedy action are classified into inferior actions, which improves the stability but decreases the optimality. To approximate the optimality on the premise of stability, we introduce an uncertainty-based trade-off to learn an adaptive e_{Q0}.

Inspired by ensemble learning, we apply a group of centralized critics $\{V_1, V_2, \cdots, V_{n_c}\}$ to estimate $\mathcal{Q}(s, \hat{\boldsymbol{u}})$, where n_c is the number of the critics. The target of $V_c(s_{t_0})$ ($c \in [1, n_c]$) is

$$\mathcal{V}_{tar}(s_{t_0}) = \begin{cases} \sum_{t=t_0}^{T} \gamma^{t-t_0} r(s_t, \boldsymbol{u}_t) & \boldsymbol{u} = \hat{\boldsymbol{u}} \\ V_c(s_{t_0}) & \boldsymbol{u} \neq \hat{\boldsymbol{u}} \end{cases} \qquad (12)$$

Assume $\{V_1(s_t), V_2(s_t), \cdots, V_{n_c}(s_t)\}$ obey the Gaussian distribution $\mathcal{N}(\bar{V}(s_t), \sigma(s_t))$, where the mean $\bar{V}(s_t)$ and variance $\sigma(s_t)$ can be acquired by maximum likelihood estimation. Instability occurs when an inferior action is misclassified into a superior action. The uncertainty of misclassification can be represented by $\sigma(s_t)$. According to

Algorithm 1 Greedy-based Value Representation

Initialize parameters θ_a and $\{\phi_1, \cdots, \phi_{c_n}\}$
Initialize replay buffer D_r and superior buffer D_s
for Iterations $i = 1, 2, \cdots$ **do**
  if test rounds **then**
    for critic $c = 1, 2, \cdots, n_c$ **do**
      Sample batch b_c from test trajectories $\boldsymbol{\tau}_{test}$
      $loss_c = \sum_{\boldsymbol{\tau}}^{b_c} \sum_t^{\boldsymbol{\tau}} [\mathcal{V}_{tar} - V_{\phi_c}(s_t)]$
    end for
  else
    Sample batch b_r randomly from D_r
    $loss_{a1} = \sum_{\boldsymbol{\tau}}^{b_r} \sum_t^{\boldsymbol{\tau}} [\mathcal{Q}_{its} - Q_{\theta_a}(\boldsymbol{u}_t, \boldsymbol{\tau}_t)]$
    Take out $\boldsymbol{\tau}_s$ with the top priority from D_s
    $loss_{a2} = \sum_t^{\boldsymbol{\tau}_s} w_{ser}(s_t) \mathbb{I}_{sup} [Q_{its} - Q_{\theta_a}(\boldsymbol{u}_t, \boldsymbol{\tau}_t)]$
    $loss_a = loss_{a1} + loss_{a2}$
    Calculate priorities for $\{b_r, \tau_s\}$ and store it to D_s
  end if
  Update θ_a and $\{\phi_1, \cdots, \phi_{c_n}\}$
end for

the 3-σ rule, we define $e_{Q0}(s_t)$ as

$$e_{Q0}(s_t) = \frac{3\sigma(s_t)}{\bar{V}(s_t)} \quad (13)$$

According to Eq.7, an action $\boldsymbol{u}_t$ is classified as the superior action when $\sum_{t=t_0}^{T} \gamma^{t-t_0} r(s_t, \boldsymbol{u}_t) > \bar{V}(s_t) + 3\sigma(s_t)$. Since $P(\bar{V}(s_t) - 3\sigma(s_t) < V(s_t) < \bar{V}(s_t) + 3\sigma(s_t)) = 0.997$, the theoretical probability of misclassification (without considerings the estimation bias of the critics) equals $P(V(s_t) > \bar{V}(s_t) + 3\sigma(s_t)) = \frac{1-0.997}{2} = 0.135\%$. As a result, an action will not be accepted as a superior action without sufficient confidence. The critics are more cautious when faced with a unfamiliar state, which prevents instability adaptively. An example is shown in Fig.3. We provide investigations of many other threshold functions in appendix J.2.

The algorithm is given in Algo.1, where $\mathbb{I}_{sup}(s_t, \boldsymbol{u}_t)$ is a indicator for the superior action, i.e., $\mathbb{I}_{sup}(s_t, \boldsymbol{u}_t) = 1$, $s.t. \sum_{t=t_0}^{T} \gamma^{t-t_0} r(s_t, \boldsymbol{u}_t) > \bar{V}(s_t) + 3\sigma(s_t)$. $w_{ser}(s_t)$ and $e_{Q0}(s_t)$ are calculated by Eq.11 and Eq.13 respectively, where $\eta_s = 1$. The priority of trajectory $\boldsymbol{\tau}$ equals to $p_{\boldsymbol{\tau}} = \sum_t^{\boldsymbol{\tau}} \mathbb{I}_{sup}(s_t, \boldsymbol{u}_t) \left[\sum_{t=t_0}^{T} \gamma^{t-t_0} r(s_t, \boldsymbol{u}_t) - \bar{V}(s_t) - 3\sigma(s_t) \right]$. For brevity, we omit the inputs of $\mathcal{V}_{tar}$, $\mathcal{Q}_{its}$ and $\mathbb{I}_{sup}$.

5. Experiments

In this section, firstly we verify our conclusion about the STNs in matrix games under difference conditions, where we also evaluate the effectiveness of GVR. Secondly, to evaluate the stability and scalability of our method, we test the performance of GVR in predator-prey tasks with extreme reward shaping and challenging tasks of StarCraft multi-agent
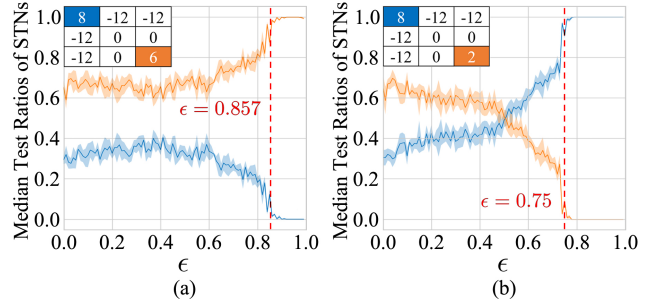

Figure 4. Median test ratios of the STNs. The pay-off matrices are shown in the upper-left of sub-graphs. In sub-graph (a), when $\epsilon = 0.4$, the ratios of the optimal STNs (blue) and the non-optimal STN (orange) approximate 0.35 and 0.65 respectively. There remains *only one* STN when $\epsilon > 0.857$, which consists with the calculated threshold (denoted by red dotted lines) from Eq.4.

challenge (SMAC) (Samvelyan et al., 2019). Finally, we design ablation studies to investigate the improvements of GVR. Our method is compared with state-of-the-art baselines including QMIX (Rashid et al., 2018), QPLEX (Wang et al., 2020), and WQMIX (Rashid et al., 2020). All results are evaluated over 5 seeds. Experimental settings and more experiments are provided in Appendix J.

5.1. One-step Matrix Game

Matrix game is a simple fully cooperative multi-agent task, where the shared rewards are defined by a pay-off matrix. In one-step matrix games, the true Q values are identical to the shared reward, which is convenient for the verification of the optimal consistency. We denote the size of a matrix by m^n, where n is the number of agents and m is the size of individual action space.
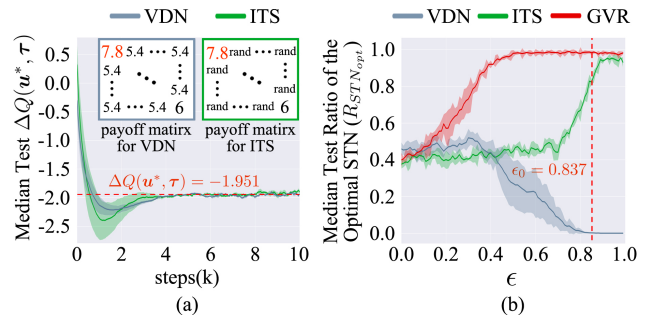

Figure 5. Evaluation of GVR in 4-agent one-step matrix games. (a) Test $\Delta Q(\boldsymbol{u}^*, \boldsymbol{\tau})$ under ITS with random rewards. (b) Median test ratio of the optimal STN ($R_{STN_{opt}}$) as ϵ grows. $R_{STN_{opt}} = 0$ denotes the optimal node is not an STN while $R_{STN_{opt}} = 1$ denotes the optimal STN is the unique STN.

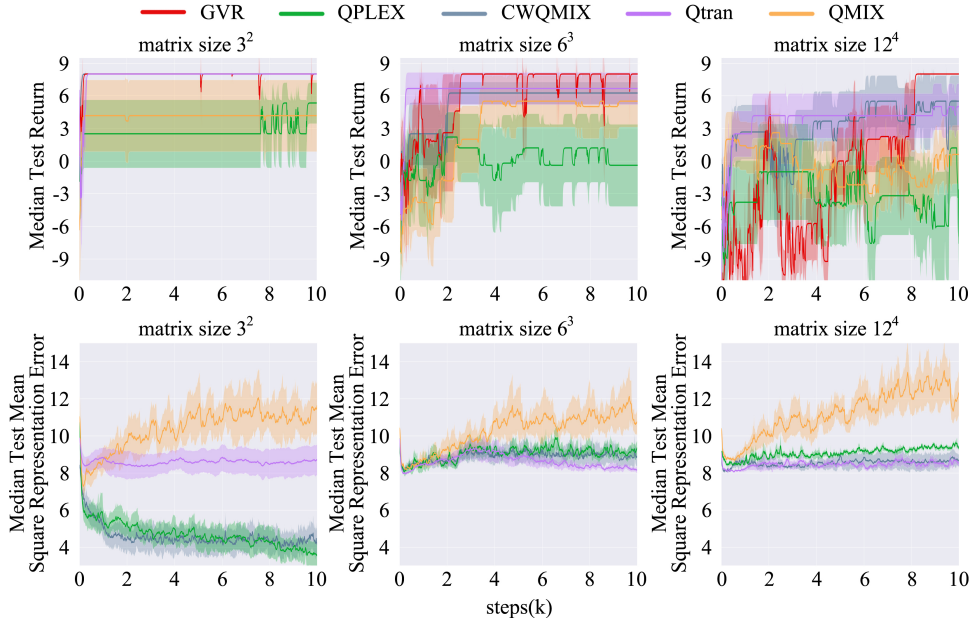Figure 6. GVR vs methods with complete representation capacity in matrix games of size 3^2, 6^3, and 12^4.

The verification of STNs under difference conditions.
We conduct experiments on two-agent one-step matrix game to verify the expression of joint Q values (i.e., Eq.4) of LVD and MVD. The experimental results are provided in Appendix C.2. Besides, we test ratios of different STNs (i.e., the probabilities of different convergence) for LVD as ϵ increasing from 0.01 to 1. As shown in Fig.4, some of the STNs disappear under large exploration, and there remains only one STN when ϵ approaches 1. However, *large exploration do not ensure the optimal consistency* because the remained STN is not necessarily the optimal STN.

Evaluation of GVR. We first verify the expression of $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ (Eq.8). Two pay-off matrices of size 3^4 are generated for VDN and ITS according to Tab.1, where $\boldsymbol{u}_s = \boldsymbol{u}^* = \{0, 0, 0, 0\}$, $\acute{\boldsymbol{u}} = \{2, 2, 2, 2\}$, $e_Q = 0.3$ and $\alpha = 0.1$. We measure $\Delta Q(\boldsymbol{u}^*, \boldsymbol{\tau}) := Q(\boldsymbol{u}^*, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ for VDN and ITS trained with corresponding matrices. As shown in Fig.5(a), the test results of $\Delta Q(\boldsymbol{u}^*, \boldsymbol{\tau})$ consists with the calculation result from Eq.8 ($\Delta Q(\boldsymbol{u}^*, \boldsymbol{\tau}) = -1.951$, denoted by the red dotted line). Besides, the test results are the same under different random seeds for ITS, which indicates the conditions in Eq.6 are independent to the true Q values of inferior actions for ITS.

To evaluate the effectiveness of our method, we compare the ratio of the optimal STN (i.e., the probability of the optimal convergence) of VDN, ITS and GVR as ϵ grows. The pay-off matrices are generated according to $\mathcal{M}(its)$ in Tab.1 over 5 seeds. At each ϵ, 100 times of independent training are executed. As shown in Fig.5(b), the optimal node is not

Table 1. Pay-off matrices generated for LVD and ITS.

$\boldsymbol{u}$	$\mathcal{M}(vdn)$	$\mathcal{M}(its)$
$\boldsymbol{u}^*$	$6(1 + e_Q)$	$6(1 + e_Q)$
$\acute{\boldsymbol{u}}$	6	6
others (inferior actions)	$6(1 - \alpha)$	random(-20,6)

an STN for VDN when $\epsilon > 0.8$ since $R_{STN_{opt}}$ becomes 0. While the optimal node is always an STN for ITS, where $R_{STN_{opt}} > 0$ always holds. Besides, the optimal STN becomes the unique STN (i.e., $R_{STN_{opt}} = 1$) for ITS under large exploration, which consists with the calculation result ($\epsilon_0 = 0.837$, denoted by the red dash line) of Eq.10. GVR fails to achieve $R_{STN_{opt}} = 1$ under a small ϵ, where the optimal action may be not explored in given training steps.

GVR vs methods with complete representation capacity.
We compare GVR to methods with complete representation capacity in matrix games of different scales. Same to $\mathcal{M}(its)$ in Tab.1, the matrix elements for inferior actions are randomly generated over 5 seeds, but the elements of $\boldsymbol{u}^*$ and $\acute{\boldsymbol{u}}$ are set to 8 and 6 respectively. As shown in Fig.6, in the matrix games of size 6^3 and 12^4, the representation errors of Qtran, QPLEX and CWQMIX do not decrease in training, which suggests they are unable to learn the complete representation within given steps. We do not measure the representation error of GVR since the target of inferior actions is modified by ITS. GVR is the only method ensuring the optimal consistency (i.e., median test return = 8) on all 3 tasks.
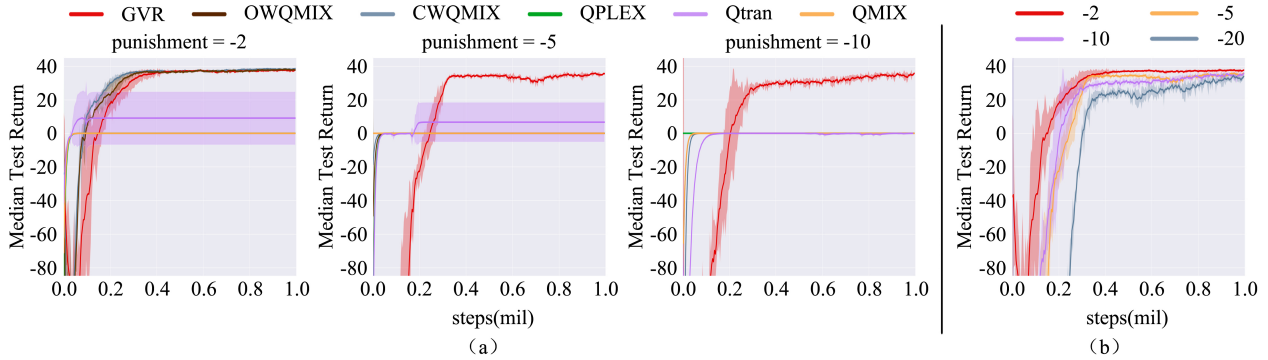
Figure 7. Experiments on predator-prey. (a) Comparison of GVR and baselines. (b) GVR under different punishments.
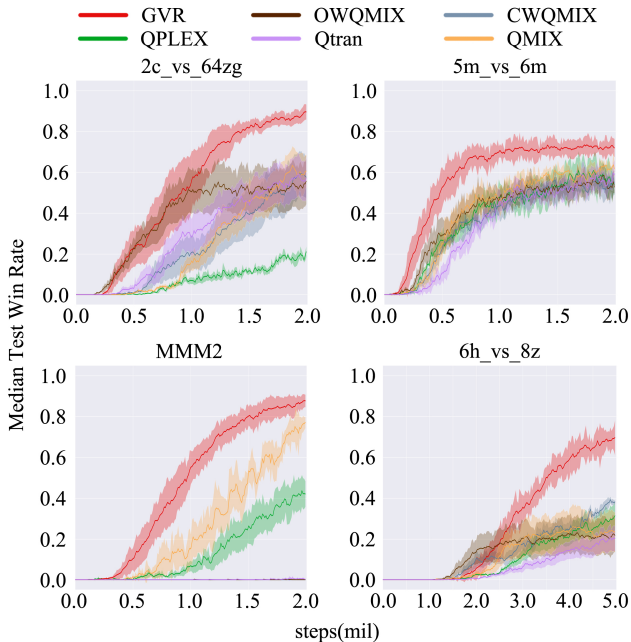

Figure 8. Comparison of GVR and baselines on SMAC.

5.2. Predator-Prey

Predator-prey is a multi-agent coordinated game, where the predators are trained to capture the preys controlled by random policies. At each time-step, the team of agents receives a positive reward instantly when more than one predators capture a same prey, but is punished with a negative reward when a prey is captured by a single agent. Otherwise, the reward is 0. As a result, the methods suffering from relative overgeneralization (RO) are unable to solve the task, where all agents tend to avoid the preys for fear of the punishment. Our experiments are carried out under 3 punishment values.

From Fig.7, VDN and QMIX fail in all tasks because both

methods suffer from the RO. QPLEX also fails in spite of the complete representation capacity. WQMIX solves the task with the punishment -2. However, the heuristic weight $\alpha = 0.1$ is insufficient to overcome the RO under large punishments. GVR is able to overcome the RO under all punishments since the true Q values of the inferior actions have little impact on the joint Q values.

5.3. StarCraft Multi-agent Challenge

We compare GVR with baselines in challenging tasks of StarCraft multi-agent challenge (SMAC). As shown in Fig.8, GVR shows the best performance. Different from predator-prey with miscoordination punishments, the reward function in SMAC is more reasonable, where the linear and monotonic value decomposition can meet the TGM principle approximately. Therefore, the algorithms with full representation expressiveness capacity (Qtran, QPLEX, WQMIX) do not perform better than QMIX due to the difficulty of complete representation.

We evaluate the effect of the adaptive trade-off, the experiment results are given in Appendix J.2. GVR achieves the trade-off through the ensemble critics, which are independent to the joint Q value function. However, since both the critics and the joint Q value function are evaluations of the policy, the independence brings conflict between two evaluations. As a result, GVR with trade-off performs worse than that without trade-off in some challenging tasks.

6. Conclusion

This paper discusses the optimality of credit assignment in value decomposition methods and proposes the optimal consistency. To achieve the optimal consistency, we introduce the TGM principle for linear and monotonic value decomposition. By deriving the expression of the joint Q value function, we draw a transition diagram. According to the diagram, we find the TGM principle can be ensured if the

optimal STN is the unique STN.

By this work, we have a deeper understating about relative overgeneralization (RO). Firstly, we prove theoretically that large exploration do not necessarily solve the RO, it depends on the reward function. Since it is challenging to explore superior actions in tasks with large joint action spaces, efficient exploration is still very important. Secondly, learning a joint Q value function with complete representation capacity is very difficult. Dispensed with complete representation capacity, a biased joint Q value function is sufficient to overcome the RO. However, the heuristic hyper-parameters is unavoidable in previous works since the STNs depend on the task-specific reward function. GVR also learns a biased joint Q value function. But different from previous works, we propose the ITS to remove the dependence of STNs on the true Q values of inferior actions. Besides, GVR achieves an adaptive trade-off between optimality and stability.

Acknowledgements

References

Böhmer, W., Kurin, V., and Whiteson, S. Deep coordination graphs. In *International Conference on Machine Learning*, pp. 980–991. PMLR, 2020.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Foerster, J. N., Assael, Y. M., De Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

Guestrin, C., Koller, D., and Parr, R. Multiagent planning with factored mdps. In *NIPS*, volume 1, pp. 1523–1530, 2001.

Gupta, T., Mahajan, A., Peng, B., Böhmer, W., and Whiteson, S. Uneven: Universal value exploration for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 3930–3941. PMLR, 2021.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.

Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. Maven: Multi-agent variational exploration. *arXiv preprint arXiv:1910.07483*, 2019.

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

Oliehoek, F. A., Spaan, M. T., and Vlassis, N. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Panait, L., Luke, S., and Wiegand, R. P. Biasing coevolutionary search for optimal multiagent behaviors. *IEEE Transactions on Evolutionary Computation*, 10(6):629–645, 2006.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:2006.10800*, 2020.

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

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896. PMLR, 2019.

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

Vorotnikov, S., Ermishin, K., Nazarova, A., and Yuschenko, A. Multi-agent robotic systems in collaborative robotics. In *International Conference on Interactive Collaborative Robotics*, pp. 270–279. Springer, 2018.

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

Wei, E., Wicke, D., Freelan, D., and Luke, S. Multiagent soft q-learning. In *2018 AAAI Spring Symposium Series*, 2018.

Wen, C., Yao, X., Wang, Y., and Tan, X. Smix (λ): Enhancing centralized value functions for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7301–7308, 2020.

Wu, T., Zhou, P., Liu, K., Yuan, Y., Wang, X., Huang, H., and Wu, D. O. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(8):8243–8256, 2020.

Yang, Y., Hao, J., Chen, G., Tang, H., Chen, Y., Hu, Y., Fan, C., and Wei, Z. Q-value path decomposition for deep multiagent reinforcement learning. In *International Conference on Machine Learning*, pp. 10706–10715. PMLR, 2020a.

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

Zhang, S. and Sutton, R. S. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*, 2017.

A. Related Work

A.1. Relative Overgeneralization and Representation Limitation

We first explain the relationship between relative overgeneralization and representation limitation. Relative overgeneralization (RO) is firstly proposed in coordination games (Panait et al., 2006), where each agent chooses an action without knowing which actions the other agents will choose. All agents receive the same rewards after performing the chosen actions. RO is related to the individual rewards, and it usually occurs in independent learning methods (e.g., independent Q learning) when an agent is punished for the miscoordination of other agents. As a result, the joint policy may converge to a suboptimal equilibrium.

In fully cooperative MARL tasks with credit assignment approaches (e.g., value decomposition), the policy of each agent is evaluated differentially, where RO depends on the credit assignment since it is related to the individual rewards. In linear value decomposition (LVD) or monotonic value decomposition (MVD), a prior linear or monotonic constraint is applied between the joint Q value function and individual utility functions, leading to the representation limitation of the joint Q value function. The representation limitation may result in poor credit assignment, which causes RO.

A.2. Value Decomposition Methods

We mainly introduce recent value decomposition methods. Value decomposition is a popular approach for credit assignment in fully cooperative MARL tasks. VDN (Sunehag et al., 2017) learns a joint Q value function based on a share reward function. In VDN, the joint Q value function is linearly factorized into individual utility functions. By contrast, QMIX (Rashid et al., 2018) substitutes the linear factorization with a monotonic factorization, where the weights and bias are produced from the global state through a mixing network. Based on QMIX, SMIX (Wen et al., 2020) replaces the TD(0) Q-learning target with a TD(λ) SARSA target. Qatten (Yang et al., 2020b) adds an attention network before the mixing network of QMIX. QPD (Yang et al., 2020a) decomposes the joint Q value function with the integrated gradient attribution technique, which directly decomposes the joint Q-values along trajectory paths to assign credits for agents. However, due to the representation limitation of the joint Q value function, these methods suffer from the RO.

Some recent works address the RO directly by completing the representation capacity of the joint Q value function. QTRAN (Son et al., 2019) learns a joint Q value function with complete representation capacity and introduces two soft regularizations to approximate the IGM principle. QPLEX (Wang et al., 2020) achieves the complete representation under IGM principle theoretically through a duelling mixing network, where the complete representation capacity is introduced by the mixing of individual advantage functions. However, as the state space and the joint action space increase exponentially as the number of agents grows, it is impractical to learn the complete representation in complicated MARL tasks, which may result in convergence difficulty and performance deterioration.

The other related works aim to prevent sub-optimal convergences by learning a biased joint Q value function. WQMIX (Rashid et al., 2020) introduces an auxiliary network to distinguish samples with low representation values. By placing a predefined low weight (α) on these samples, WQMIX learns a biased joint Q value function which focuses on the representation of actions with good performance. According to our analysis in Appendix F.1, a relatively higher weight on the superior samples helps to eliminate non-optimal STNs under ITS. However, WQMIX can not ensure the optimal consistency for *two reasons*. (1) According to Eq.25 (Appendix C.1), the joint Q value function under MVD depends on the true Q values of all actions in the whole joint action space, which are unavailable and task-specific. As a result, a heuristic weight has to be adopted in WQMIX. (2) According to Tab.4 (Appendix F.2), the required weight ($\alpha = \frac{1}{w_0}$) in WQMIX is very small under ITS, which introduces instability in training. Although WQMIX eliminates all non-optimal STNs theoretically when $\alpha = 0$, the optimal STN is unstable (i.e., the optimal node can not keep self-transition) because the joint Q values of actions with low representation values have not been trained.

MAVEN (Mahajan et al., 2019) focuses on the poor exploration that arises from the representation limitation and introduces a latent space for hierarchical control, which achieves temporally extended exploration. UneVEn (Gupta et al., 2021) solves a target task by learning a set of related tasks simultaneously with a linear decomposition of universal successor features, which improves the joint exploration. Both methods improve the joint exploration, which helps to eliminate non-optimal STNs. However, they can not ensure the optimal consistency for *two reasons*. (1) According to Fig.4 (Section 5.1), improving exploration does not necessary ensure the optimal consistency. Instead, it depends on the reward function. (2) According to Eq.9 in Section 4.1, to eliminate non-optimal STNs, the required exploration rate approximates 1 under ITS, which is inapplicable in tasks with long episodes.

Table 2. Comparison between GVR and related works.

METHODS	ENSURES IGM PRINCIPLE	ENSURES TGM PRINCIPLE	TRADE-OFF BETWEEN OPTIMALITY AND STABILITY
IQL	×	×	×
VDN	✓	×	×
QMIX	✓	×	×
SMIX	✓	×	×
QATTEN	✓	×	×
QDP	×	×	×
QTRAN	×	✓	×
MAVEN	✓	×	×
UNEVEN	✓	×	×
WQMIX	✓	×	×
QPLEX	✓	✓	×
GVR(OURS)	✓	✓	✓

Besides, as discussed in Appendix H.2, excessive pursuit for optimality decreases the stability. Without trade-off between optimality and stability, the methods which approximate the optimal consistency (e.g., WQMIX under extreme small weights and QPLEX) suffer from the risk of instability.

B. Joint Q Value Function of LVD in Awo-agent Cooperation

Consider a two-agent fully cooperative task with LVD. The joint Q value function $Q(u_i^1, u_j^2, \boldsymbol{\tau})$ is linearly factorized into two utility functions as

$$Q(u_i^1, u_j^2, \boldsymbol{\tau}) = \mathcal{U}^1(u_i^1, \tau^1) + \mathcal{U}^2(u_j^2, \tau^2) \tag{14}$$

where $u_i^1, u_j^2 \in \{u_1, \cdots, u_m\}$ are the actions of agents 1 and 2, respectively. $\{u_1, \cdots, u_m\}$ is the discrete individual action space. Specially, the greedy actions of two agents are denoted by $\acute{u}^1$ and $\acute{u}^2$. For briefness, we denote $\mathcal{U}^a(u_i^a, \tau^a)$ and the true Q value function $\mathcal{Q}(s, u_i^1, u_j^2)$ with $\mathcal{U}_i^a(a \in \{1, 2\})$ and $\mathcal{Q}_{ij}$, respectively. Since the utility functions are trained to approximate different true Q values in different combinations, under $\epsilon-$greedy visitation, we have

$$\mathcal{U}_i^1 = \frac{\epsilon}{m} \sum_{k=1}^{m} (\mathcal{Q}_{ik} - \mathcal{U}_k^2) + (1 - \epsilon)(\mathcal{Q}_{i\acute{j}} - \mathcal{U}_{\acute{j}}^2)$$

$$\mathcal{U}_j^2 = \frac{\epsilon}{m} \sum_{k=1}^{m} (\mathcal{Q}_{kj} - \mathcal{U}_k^1) + (1 - \epsilon)(\mathcal{Q}_{\acute{i}j} - \mathcal{U}_{\acute{i}}^1) \tag{15}$$

The sum of two agents' utility functions over all actions equals to

$$\sum_{i=1}^{m} \mathcal{U}_i^1 + \sum_{j=1}^{m} \mathcal{U}_j^2 = \frac{\epsilon}{m} \left[\sum_{i=1}^{m} \sum_{k=1}^{m} \mathcal{Q}_{ik} + \sum_{j=1}^{m} \sum_{k=1}^{m} \mathcal{Q}_{kj} - m \sum_{k=1}^{m} (\mathcal{U}_k^1 + \mathcal{U}_k^2) \right] + (1 - \epsilon) \left[\sum_{i=1}^{n} (\mathcal{Q}_{i\acute{j}} + \mathcal{Q}_{\acute{i}j}) - m(\mathcal{U}_{\acute{i}}^1 + \mathcal{U}_{\acute{j}}^2) \right] \tag{16}$$

Notice that $\mathcal{U}_{\acute{i}}^1 + \mathcal{U}_{\acute{j}}^2 = Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau})$, and $\sum_{i=1}^{m} \sum_{k=1}^{m} \mathcal{Q}_{ik} = \sum_{j=1}^{m} \sum_{k=1}^{m} \mathcal{Q}_{kj} = \sum_{i=1}^{m} \sum_{j=1}^{m} \mathcal{Q}_{ij}$, we have

$$\sum_{k=1}^{m} (\mathcal{U}_k^1 + \mathcal{U}_k^2) = \frac{2\epsilon}{m(1+\epsilon)} \sum_{i=1}^{m} \sum_{j=1}^{m} \mathcal{Q}_{ij} + \frac{1-\epsilon}{1+\epsilon} \sum_{k=1}^{m} (\mathcal{Q}_{ik} + \mathcal{Q}_{k\acute{j}}) - \frac{m(1-\epsilon)}{1+\epsilon} Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau}) \tag{17}$$

According to Eq.15 and Eq.17, for $\forall i, j \in [1, m]$, the joint Q value function equals to

$$
\begin{aligned}
Q(u_i^1, u_j^2, \boldsymbol{\tau}) =& \mathcal{U}_i^1 + \mathcal{U}_j^2 = \frac{\epsilon}{m} \left[\sum_{k=1}^m (\mathcal{Q}_{ik} + \mathcal{Q}_{kj}) - \sum_{k=1}^m (\mathcal{U}_k^1 + \mathcal{U}_k^2) \right] + (1 - \epsilon)(\mathcal{Q}_{i\acute{j}} + \mathcal{Q}_{\acute{i}j} - Q_{\acute{i}\acute{j}}) \\
=& \frac{\epsilon}{m} \sum_{k=1}^m (\mathcal{Q}_{ik} + \mathcal{Q}_{kj}) + (1 - \epsilon)(\mathcal{Q}_{i\acute{j}} + \mathcal{Q}_{\acute{i}j}) - \frac{2\epsilon^2}{m^2(1+\epsilon)} \sum_{i=1}^m \sum_{j=1}^m \mathcal{Q}_{ij} \\
& - \frac{\epsilon(1-\epsilon)}{m(1+\epsilon)} \sum_{k=1}^m (\mathcal{Q}_{\acute{i}k} + \mathcal{Q}_{k\acute{j}}) - \frac{1-\epsilon}{1+\epsilon} Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau})
\end{aligned}
\tag{18}
$$

Notice that $Q(u_i^1, u_j^2, \boldsymbol{\tau})$ is related to the joint greedy Q value $Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau})$. To remove it, substituting $u_i^1 = \acute{u}^1$ and $u_j^2 = \acute{u}^2$ into Eq.18, we have

$$
Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau}) = \frac{\epsilon^2}{m} \sum_{k=1}^m (\mathcal{Q}_{\acute{i}k} + \mathcal{Q}_{k\acute{j}}) - \frac{\epsilon^2}{m^2} \sum_{i=1}^m \sum_{j=1}^m \mathcal{Q}_{ij} + (1 - \epsilon^2) \mathcal{Q}_{\acute{i}\acute{j}}
\tag{19}
$$

Substituting Eq.19 into Eq.18, $Q(u_i^1, u_j^2, \boldsymbol{\tau})$ can be represented by true Q values as

$$
\begin{aligned}
Q(u_i^1, u_j^2, \boldsymbol{\tau}) =& \frac{\epsilon}{m} \sum_{k=1}^m (\mathcal{Q}_{ik} + \mathcal{Q}_{kj}) + (1 - \epsilon)(\mathcal{Q}_{i\acute{j}} + \mathcal{Q}_{\acute{i}j}) - \frac{\epsilon^2}{m^2} \sum_{i=1}^m \sum_{j=1}^m \mathcal{Q}_{ij} \\
& - \frac{\epsilon(1-\epsilon)}{m} \sum_{k=1}^m (\mathcal{Q}_{\acute{i}k} + \mathcal{Q}_{k\acute{j}}) - (1 - \epsilon)^2 \mathcal{Q}_{\acute{i}\acute{j}}
\end{aligned}
\tag{20}
$$

C. Joint Q Value Function of MVD in Awo-agent Cooperation

C.1. Derivation

For two-agent monotonic value decomposition, the joint Q value function is decomposed as $Q_{ij} = \omega_1(s)\mathcal{U}_i^1 + \omega_2(s)\mathcal{U}_j^2 + V(s)$, where $V(s)$ is a bias. $\omega_1(s)$ and $\omega_2(s)$ are the coefficients of $\mathcal{U}_i^1$ and $\mathcal{U}_j^2$, respectively. For brevity, we omit their inputs. Referring to Eq.15, the individual utility functions with coefficients equal to

$$
\begin{aligned}
\omega_1 \mathcal{U}_i^1 =& \frac{\epsilon}{m} \sum_{k=1}^m \left[\mathcal{Q}_{ik} - \omega_2 \mathcal{U}_k^2 - V \right] + (1 - \epsilon) \left[\mathcal{Q}_{i\acute{j}} - \omega_2 \mathcal{U}_{\acute{j}}^2 - V \right] \\
=& \frac{\epsilon}{m} \sum_{k=1}^m \left[\mathcal{Q}_{ik} - \omega_2 \mathcal{U}_k^2 \right] + (1 - \epsilon) \left[\mathcal{Q}_{i\acute{j}} - \omega_2 \mathcal{U}_{\acute{j}}^2 \right] - V \\
\omega_2 \mathcal{U}_j^2 =& \frac{\epsilon}{m} \sum_{k=1}^m \left[\mathcal{Q}_{kj} - \omega_1 \mathcal{U}_k^1 - V \right] + (1 - \epsilon) \left[\mathcal{Q}_{\acute{i}j} - \omega_1 \mathcal{U}_{\acute{i}}^1 - V \right] \\
=& \frac{\epsilon}{m} \sum_{k=1}^m \left[\mathcal{Q}_{ik} - \omega_2 \mathcal{U}_k^2 \right] + (1 - \epsilon) \left[\mathcal{Q}_{i\acute{j}} - \omega_2 \mathcal{U}_{\acute{j}}^2 \right] - V
\end{aligned}
\tag{21}
$$

Notice $\omega_1(s)$, $\omega_2(s)$ and $V(s)$ are independent of the actions. Referring to the derivation of Eq.17, we have

$$
\sum_{k=1}^m (\omega_1 \mathcal{U}_k^1 + \omega_2 \mathcal{U}_k^2) = \frac{2\epsilon}{m(1+\epsilon)} \sum_{i=1}^m \sum_{j=1}^m \mathcal{Q}_{ij} + \frac{1-\epsilon}{1+\epsilon} \sum_{k=1}^m (\mathcal{Q}_{\acute{i}k} + \mathcal{Q}_{k\acute{j}}) - \frac{m(1-\epsilon)}{1+\epsilon} Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau}) - mV
\tag{22}
$$

According to Eq.21 and Eq.22, we have

$$
\begin{aligned}
Q(u_i^1, u_j^2, \boldsymbol{\tau}) =& \frac{\epsilon}{m} \sum_{k=1}^m (\mathcal{Q}_{ik} + \mathcal{Q}_{kj}) + (1 - \epsilon)(\mathcal{Q}_{i\acute{j}} + \mathcal{Q}_{\acute{i}j}) - \frac{2\epsilon^2}{m^2(1+\epsilon)} \sum_{i=1}^m \sum_{j=1}^m \mathcal{Q}_{ij} \\
& - \frac{1-\epsilon}{1+\epsilon} Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau}) - \frac{\epsilon(1-\epsilon)}{m(1+\epsilon)} \sum_{k=1}^m (\mathcal{Q}_{\acute{i}k} + \mathcal{Q}_{k\acute{j}})
\end{aligned}
\tag{23}
$$

In order to remove $Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau})$ from Eq.23, let $u_i^1 = \acute{u}^1$ and $u_j^2 = \acute{u}^2$.

$$Q(\acute{u}^1, \acute{u}^2, \boldsymbol{\tau}) = \frac{\epsilon^2}{m}\sum_{k=1}^{m}(\mathcal{Q}_{ik} + \mathcal{Q}_{k\acute{j}}) - \frac{\epsilon^2}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathcal{Q}_{ij} + (1-\epsilon^2)\mathcal{Q}_{i\acute{j}} \tag{24}$$

Substituting Eq.24 into Eq.23, we have

$$\begin{aligned}
Q(u_i^1, u_j^2, \boldsymbol{\tau}) =& \frac{\epsilon}{m}\sum_{k=1}^{m}(\mathcal{Q}_{ik} + \mathcal{Q}_{kj}) + (1-\epsilon)(\mathcal{Q}_{ij} + \mathcal{Q}_{i\acute{j}}) - \frac{\epsilon^2}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathcal{Q}_{ij} \\
& - \frac{\epsilon(1-\epsilon)}{m}\sum_{k=1}^{m}(\mathcal{Q}_{ik} + \mathcal{Q}_{k\acute{j}}) - (1-\epsilon)^2\mathcal{Q}_{i\acute{j}}
\end{aligned} \tag{25}$$

Eq.25 is the same as Eq.20, which indicates LVD and MVD share *the same expression* of the joint Q value function.

C.2. Verification

We verify the derived joint Q value function (i.e., Eq.20) for LVD and MVD in a two-agent matrix game, where the payoff matrix is shown in Tab.3(a). Since the episode length is 1, an mlp shared by two agents is adopted as the policy network. The policy network is trained for 500 iterations (100 episodes per iteration) over 5 seeds under $\epsilon = 0.2$. As shown in Tab.3(b) and Tab.3(c), there are two STNs. The test results of joint Q values for LVD and MVD indicates LVD and MVD share the same expression of the joint Q value function. Besides, we measure the square error of joint Q values between test ($Q_{test}(\boldsymbol{u}, \boldsymbol{\tau})$) and calculation ($Q_{cal}(\boldsymbol{u}, \boldsymbol{\tau})$), i.e., $\sum_{\boldsymbol{u}}^{U}\left[Q_{cal}(\boldsymbol{u}, \boldsymbol{\tau}) - Q_{test}(\boldsymbol{u}, \boldsymbol{\tau})\right]^2$. The result is shown in Fig.9, where MVD converges faster than LVD.

Figure 9. Square error of $Q(\boldsymbol{u}, \boldsymbol{\tau})$ between test and calculation.

Table 3. Verification of the joint Q value function (Eq.20) for two-agent LVD and MVD. (a) The pay-off matrix. (b),(c) Comparison between test and calculation joint Q values for LVD and MVD on two STNs. We mark the calculation result, test result of LVD and MVD with (C), (L) and (M) respectively. The greedy policies are marked with pink backgrounds.

8	-12	-12
-12	0	0
-12	0	6

(a)

7.40(C) 7.38±0.02(L) 7.39±0.07(M)	-8.33(C) -8.27±0.12(L) -8.56±0.06(M)	-7.93(C) -7.86±0.13(L) -7.88±0.08(M)
-8.33(C) -8.33±0.18(L) -8.18±0.12(M)	-24.06(C) -23.87±0.09(L) -24.51±0.06(M)	-23.66(C) -23.56±0.17(L) -23.41±0.17(M)
-7.93(C) -7.93±0.09(L) -8.08±0.14(M)	-23.66(C) -23.53±0.12(L) -24.44±0.17(M)	-23.26(C) -23.19±0.20(L) -23.36±0.23(M)

(b)

-24.38(C) -24.34±0.25(L) -23.98±0.18(M)	-14.52(C) -14.52±0.11(L) -14.34±0.08(M)	-9.32(C) -9.43±0.15(L) -9.20±0.05(M)
-14.52(C) -14.47±0.18(L) -14.23±0.09(M)	-4.65(C) -4.65±0.11(L) -4.59±0.06(M)	0.55(C) 0.54±0.08(L) 0.55±0.05(M)
-9.32(C) -9.28±0.21(L) -9.07±0.14(M)	0.55(C) 0.56±0.12(L) 0.57±0.06(M)	5.75(C) 5.75±0.09(L) 5.72±0.05(M)

(c)

D. Derivation of $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ under ITS

D.1. Examples of the ITS Target

Examples of the ITS target under different greedy actions are shown in Fig.10.

$\mathcal{Q}_{its}(s, \boldsymbol{u})$	(a)	(b)	(c)
$\acute{\boldsymbol{u}}$	$\{0, 0\}$	$\{1, 1\}$	$\{2, 2\}$
$Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$	7.8	-0.2	5.8
$\mathcal{Q}_{its}(s, \boldsymbol{u})$ for inferior actions: $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \alpha\lvert Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})\rvert$	7.02	-0.22	5.22

$\mathcal{Q}(s, \boldsymbol{u})$ table:

8	-12	-12
-12	0	0
-12	0	6

- ☐ greedy action
- ☐ inferior action
- ☐ superior action

(a)

8	7.02	7.02
7.02	7.02	7.02
7.02	7.02	7.02

(b)

8	-0.22	-0.22
-0.22	0	0
-0.22	0	6

(c)

8	5.22	5.22
5.22	5.22	5.22
5.22	5.22	6

Figure 10. Examples of the ITS target under different $\acute{\boldsymbol{u}}$, where $e_{Q0} = 0.1$ and $\alpha = 0.1$. The true Q values are given in the yellow table.

D.2. Derivation 1

Given the greedy action $\{\acute{u}^1, \cdots, \acute{u}^n\} = \acute{\boldsymbol{u}}$ and any other action $\{u_s^1, \cdots, u_s^n\} = \boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$, here we regard all actions except $\boldsymbol{u}_s$ and $\acute{\boldsymbol{u}}$ as inferior actions. Discussions under multiple superior actions is provided in Appendix H.1. We first consider the hardest exploration case, where $\forall a \in [1, n]$, $u_s^a \neq \acute{u}^a$. The derivation under non-hardest exploration cases is provided in Appendix D.4. For simplicity, assuming $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) > 0$, where $\mathcal{Q}_{its}(s, \boldsymbol{u}) = (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ for inferior actions, the utility function of individual action $u_s^a (a \in [1, n])$ is consist of two parts

$$\mathcal{U}^a(u_s^a, \tau^a) = (1 - \eta_1)\left[(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_{k}^{m^{n-1}-1}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a) - p(\boldsymbol{u}_s)}\sum_{i}^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right]\right]$$
$$+ \eta_1\left[\mathcal{Q}_{its}(s, \boldsymbol{u}_s) - \sum_{i}^{-a}\mathcal{U}^i(u_s^i, \tau^i)\right] \quad (26)$$

where $\eta_1 = \left(\frac{\epsilon}{m}\right)^{n-1}$, and $-a$ represents the collection of all agents except agent a. η_1 and $1 - \eta_1$ are the proportions of $\boldsymbol{u}_s$ and inferior actions in $\sum_k^{m^{n-1}}\{u_s^a, u_k^{-a}\}$ respectively, where $\sum_k^{m^{n-1}}\{u_s^a, u_k^{-a}\}$ denotes all joint actions containing u_s^a, and $p(u_s^a, u_k^{-a})$ is the corresponding probability of each joint action. The superscript of the first $\sum$ in Eq.26 is $m^{n-1} - 1$ because $\boldsymbol{u}_s$ is excluded. $p(u_s^a) - p(\boldsymbol{u}_s) = \sum_k^{m^{n-1}} p(u_s^a, u_k^{-a})$ is the normalization coefficient, where $\sum_k^{m^{n-1}} p(u_s^a, u_k^{-a}) = \frac{\epsilon}{m} - \left(\frac{\epsilon}{m}\right)^n$. Notice

$$(1 - \eta_1)\sum_k^{m^{n-1}-1}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a) - p(\boldsymbol{u}_s)}\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right] + \eta_1\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i) = \sum_k^{m^{n-1}}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a)}\sum_i^{-a}\mathcal{U}^i(u_s^i, \tau^i)\right] \quad (27)$$

Therefore,

$$\mathcal{U}^a(u_s^a, \tau^a) = (1 - \eta_1)(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_k^{m^{n-1}}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a)}\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right] + \eta_1\mathcal{Q}_{its}(s, \boldsymbol{u}_s) \quad (28)$$

The joint Q value function $Q(\boldsymbol{u}_s, \boldsymbol{\tau})$ can be acquired

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) = \sum_{a=1}^{n}\mathcal{U}^a(u_s^a, \tau^a) = n(1 - \eta_1)(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_{a=1}^{n}\sum_k^{m^{n-1}}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a)}\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right] + n\eta_1\mathcal{Q}_{its}(s, \boldsymbol{u}_s) \quad (29)$$

Similarly, for the greedy action $\acute{\boldsymbol{u}}$, we have

$$\mathcal{U}^a(\acute{u}^a, \tau^a) = (1 - \eta_2)(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_{k}^{m^{n-1}} \left[\frac{p(\acute{u}^a, u_k^{-a})}{p(\acute{u}^a)} \sum_i^{-a} \mathcal{U}^i(u_k^i, \tau^i) \right] + \eta_2 \mathcal{Q}(s, \acute{\boldsymbol{u}}) \tag{30}$$

where $\eta_2 = (1 - \epsilon + \frac{\epsilon}{m})^{n-1}$. As a result,

$$Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) = \sum_{a=1}^n \mathcal{U}^a(\acute{u}^a, \tau^a) = n(1 - \eta_2)(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_{a=1}^n \sum_k^{m^{n-1}} \left[\frac{p(\acute{u}^a, u_k^{-a})}{p(\acute{u}^a)} \sum_i^{-a} \mathcal{U}^i(u_k^i, \tau^i) \right] + n\eta_2 \mathcal{Q}(s, \acute{\boldsymbol{u}}) \tag{31}$$

Notice $\frac{p(u^a, u^{-a})}{p(u^a)}$ is independent of action u^a for decentralized execution, therefore $\frac{p(\acute{u}^a, u_k^{-a})}{p(\acute{u}^a)} = \frac{p(u_s^a, u_k^{-a})}{p(u_s^a)}$. Let $\mathcal{Q}(s, \boldsymbol{u}_s) = (1 + e_Q)\mathcal{Q}(s, \acute{\boldsymbol{u}})$,according to Eq.29 and Eq.31, we have

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) = n(\eta_1 - \eta_2) \left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \right] + n\eta_1 e_Q \mathcal{Q}(s, \acute{\boldsymbol{u}}) \tag{32}$$

For monotonic value decomposition, Eq.32 also holds since $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ and $Q(\boldsymbol{u}_s, \boldsymbol{\tau})$ do not change. Verification of Eq.32 is provided in the experimental part.

D.3. Derivation 2

Given the greedy action $\{\acute{u}^1, \cdots, \acute{u}^n\} = \acute{\boldsymbol{u}}$ and any other action $\{u_s^1, \cdots, u_s^n\} = \boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$, here we regard all actions except $\boldsymbol{u}_s$ and $\acute{\boldsymbol{u}}$ as inferior actions. Discussions under multiple superior actions is provided in Appendix H.1. We first consider the hardest exploration case, where $\forall a \in [1, n]$, $u_s^a \neq \acute{u}^a$. The derivation under non-hardest exploration cases is provided in Appendix D.4. For simplicity, assuming $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) > 0$, where $\mathcal{Q}_{its}(s, \boldsymbol{u}) = (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ for inferior actions, the utility function of u_s^a equals to

$$\begin{aligned}
\mathcal{U}_{u_s^a}^a &= \left(\frac{\epsilon}{m}\right)^{n-1} \left[C_{n-1}^0 (m^{n-1} - 1)(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + \mathcal{Q}(s, \boldsymbol{u}_s) + f_1(\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a) \right] \\
&\quad + \cdots + \left(\frac{\epsilon}{m}\right)^{n-t} (1 - \epsilon)^{t-1} \left[C_{n-1}^{t-1} m^{n-t}(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + f_t(\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a) \right] + \cdots \\
&\quad + (1 - \epsilon)^{n-1} \left[C_{n-1}^{n-1}(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + f_n(\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a) \right] \\
&= (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + \left(\frac{\epsilon}{m}\right)^{n-1} \left[\mathcal{Q}(s, \boldsymbol{u}_s) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \right] + f_{total}(\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a)
\end{aligned} \tag{33}$$

where $-a$ represents the collection of all agents except agent a. $f_t(t \in [1, n])$ and f_{total} are mappings from $\{\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a\}$ to $\mathbb{R}$. The joint Q value of $\boldsymbol{u}_s$ equals to

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) = \sum_{a=1}^n \mathcal{U}_{u_s^a}^a = n(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\left(\frac{\epsilon}{m}\right)^{n-1} \left[\mathcal{Q}(s, \boldsymbol{u}_s) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \right] + nf_{total}(\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a) \tag{34}$$

Next we calculate the joint Q value of $\acute{\boldsymbol{u}}$. The utility function of the individual greedy action $\acute{u}^a (a \in [1, n])$ equals to

$$\mathcal{U}_{\acute{u}^a}^a = (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + (1 - \epsilon + \frac{\epsilon}{m})^{n-1} \left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \right] + f_{total}(\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a) \tag{35}$$

The joint Q value of $\acute{\boldsymbol{u}}$ equals to

$$\begin{aligned}
Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) &= \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a \\
&= n(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n(1 - \epsilon + \frac{\epsilon}{m})^{n-1} \left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \right] + nf_{total}(\sum_o^{-a}\sum_{i=1}^m \mathcal{U}_{u_i^o}^o, \sum_{a=1}^n \mathcal{U}_{\acute{u}^a}^a)
\end{aligned} \tag{36}$$

Let $\eta_1 = (\frac{\epsilon}{m})^{n-1}$, $\eta_2 = (1 - \epsilon + \frac{\epsilon}{m})^{n-1}$, and $\mathcal{Q}(s, \boldsymbol{u}_s) = (1 + e_Q)\mathcal{Q}(s, \acute{\boldsymbol{u}})$, we have

$$
\begin{aligned}
Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) &= n\eta_1 \left[(1 + e_Q)\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})\right] - n\eta_2 \left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})\right] \\
&= n(\eta_1 - \eta_2)\left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})\right] + n\eta_1 e_Q \mathcal{Q}(s, \acute{\boldsymbol{u}})
\end{aligned}
\tag{37}
$$

which *consists with* Eq.32 in Derivation 1.

D.4. Non-hardest Exploration Cases

In Proof 1 and Proof 2, we only consider the hardest exploration case, where $\forall a \in [1, n]$, $u_s^a \neq \acute{u}^a$. In this subsection, we derive $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ in general cases where

$$
u_s^a \begin{cases} = \acute{u}^a & a \in \mathcal{L} \\ \neq \acute{u}^a & others \end{cases}
\tag{38}
$$

$\mathcal{L} \in \{1, \cdots, n\}$ and $\mathcal{L} \neq \emptyset$. Assuming $\mathcal{Q}(s, \acute{\boldsymbol{u}}) > 0$, i.e., $\mathcal{Q}_{its}(s, \boldsymbol{u}) = (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ for inferior samples. When $a \in \mathcal{L}$, the utility function of individual action u_s^a is consist of three parts

$$
\begin{aligned}
\mathcal{U}^a(u_s^a, \tau^a) =&\eta_1' \left[\mathcal{Q}_{its}(s, \boldsymbol{u}_s) - \sum_i^{-a} \mathcal{U}^i(u_s^i, \tau^i)\right] + \eta_2 \left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - \sum_i^{-a} \mathcal{U}^i(\acute{u}^i, \tau^i)\right] \\
&+ (1 - \eta_1' - \eta_2)\left[(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_k^{m^{n-1}-2} \left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a) - p(\boldsymbol{u}_s)} \sum_i^{-a} \mathcal{U}^i(u_k^i, \tau^i)\right]\right]
\end{aligned}
\tag{39}
$$

where $\eta_1' = (\frac{\epsilon}{m})^{n-l}(1 - \epsilon - \frac{\epsilon}{m})^{l-1}$, $\eta_2 = (1 - \epsilon - \frac{\epsilon}{m})^{n-1}$. When $a \notin \mathcal{L}$, according to Eq.28, we have

$$
\mathcal{U}^a(u_s^a, \tau^a) = (1 - \eta_1')(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_k^{m^{n-1}} \left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a)} \sum_i^{-a} \mathcal{U}^i(u_k^i, \tau^i)\right] + \eta_1' \mathcal{Q}_{its}(s, \boldsymbol{u}_s)
\tag{40}
$$

As a result,

$$
\mathcal{U}^a(u_s^a, \tau^a) = \begin{cases} Eq.39 & a \in \mathcal{L} \\ Eq.40 & others \end{cases} \qquad \mathcal{U}^a(\acute{u}^a, \tau^a) = \begin{cases} Eq.39 & a \in \mathcal{L} \\ Eq.30 & others \end{cases}
\tag{41}
$$

Therefore,

$$
\begin{aligned}
Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) &= \sum_a^{[1,n]-\mathcal{L}} \left[\mathcal{U}^a(u_s^a, \tau^a) - \mathcal{U}^a(\acute{u}^a, \tau^a)\right] \\
&= (n - l)(\eta_1' - \eta_2)\left[\mathcal{Q}(s, \acute{\boldsymbol{u}}) - (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})\right] + (n - l)\eta_1' e_Q \mathcal{Q}(s, \acute{\boldsymbol{u}})
\end{aligned}
\tag{42}
$$

E. STNs under ITS

E.1. The Optimal STN

Suppose $\acute{\boldsymbol{u}}$ equals to the optimal action, i.e., $\acute{\boldsymbol{u}} = \boldsymbol{u}^*$. Notice the sum of coefficients of all true Q values in Eq.20 equals

$$
2m \cdot \frac{\epsilon}{m} + 2(1 - \epsilon) - m^2 \cdot \frac{\epsilon^2}{m^2} - 2m \cdot \frac{\epsilon(1 - \epsilon)}{m} - (1 - \epsilon)^2 = 1
\tag{43}
$$

In this case, there are only two kinds of actions: greedy action and inferior actions. Inspired by Eq.43, $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ can be written as

$$
Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) = w\mathcal{Q}(s, \acute{\boldsymbol{u}}) + (1 - w)\mathcal{Q}_{its}(s, \boldsymbol{u}_s)
\tag{44}
$$

where $\mathcal{Q}_{its}(s, \boldsymbol{u}_s) = (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ is the ITS target of inferior actions. Therefore,

$$
\frac{\mathcal{Q}(s, \acute{\boldsymbol{u}})}{Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})} = 1 + \alpha\frac{1 - w}{w}
\tag{45}
$$

We first consider the hardest exploration case. Substituting Eq.45 into Eq.32, we have

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) = n(\eta_1 - \eta_2)\frac{\alpha}{w}Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\eta_1 e_Q \mathcal{Q}(s, \acute{\boldsymbol{u}}) \tag{46}$$

Since $\acute{\boldsymbol{u}} = \boldsymbol{u}^*$, $\forall \boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$, $\mathcal{Q}(s, \boldsymbol{u}_s) < \mathcal{Q}(s, \acute{\boldsymbol{u}})$ holds. Therefore, $e_Q < 0$. Notice $\eta_1 < \eta_2$, we have $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) < 0$, which indicates $Condition$ 1 (Eq.5, Section 3.2) *always holds under ITS*, i.e., ITS turn the optimal node into an STN. For non-hardest exploration case, the conclusion also holds.

E.2. The Non-optimal STNs

Suppose $\acute{\boldsymbol{u}}$ is a non-optimal action, $\exists \boldsymbol{u}_s$ such that $\mathcal{Q}(s, \boldsymbol{u}_s) > \mathcal{Q}(s, \acute{\boldsymbol{u}})$. According to $Conidtion$ 2 (Eq.5, Section 3.2), to ensure this non-optimal point is not an STN, let $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) > Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ and assume $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \approx \mathcal{Q}(s, \acute{\boldsymbol{u}})$ (this assumption is quite accurate under ITS, as verified in Appendix F.2), for the hardest exploration case we have

$$\frac{\eta_1}{\eta_2} > \frac{\alpha}{\alpha + e_{Q0}} \tag{47}$$

where $e_{Q0} \in [0, \infty)$ is a hyper-parameter that defines the minimum gap of the true Q values between the superior and greedy actions. Therefore, *the non-optimal STNs can be eliminated by raising $\frac{\eta_1}{\eta_2}$ under ITS*.

For the non-hardest exploration case, η_1 in Eq.47 is replaced by η_1'. Since $\frac{\eta_1}{\eta_2} > \frac{\eta_1'}{\eta_2}$, we only need to consider the hardest exploration case.

F. STNs under ITS with Superior Sample Weight

F.1. Derivation of $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$

Given the greedy action $\{\acute{u}^1, \cdots, \acute{u}^n\} = \acute{\boldsymbol{u}}$ and a superior action $\{u_s^1, \cdots, u_s^n\} = \boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$, we have $\mathcal{Q}(s, \boldsymbol{u}_s) > \mathcal{Q}(s, \acute{\boldsymbol{u}})$. Here we regard all actions except $\boldsymbol{u}_s$ and $\acute{\boldsymbol{u}}$ as inferior actions. Discussions under multiple superior actions is provided in Appendix H.1. We first consider the hardest exploration case. For simplicity, assuming $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) > 0$, where $\mathcal{Q}_{its}(s, \boldsymbol{u}) = (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ for inferior actions. By applying a weight w on the superior action, the utility function of the individual action $u_s^a (a \in [1, n])$ consists of two parts

$$
\begin{aligned}
\mathcal{U}^a(u_s^a, \tau^a) =& (1 - \eta_{1,w})\left[(1-\alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_k^{m^{n-1}-1}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a) - p(\boldsymbol{u}_s)}\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right]\right] \\
&+ \eta_{1,w}\left[\mathcal{Q}_{its}(s, \boldsymbol{u}_s) - \sum_i^{-a}\mathcal{U}^i(u_s^i, \tau^i)\right]
\end{aligned}
\tag{48}
$$

where $\eta_{1,w} = \frac{w\eta_1}{1+(w-1)\eta_1}$, $\eta_1 = (\frac{\epsilon}{m})^{n-1}$. Please refer to Appendix D.2 for more details about the notations. According to Eq.27, we have

$$\sum_k^{m^{n-1}-1}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a) - p(\boldsymbol{u}_s)}\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right] = \frac{1}{(1-\eta_1)}\sum_k^{m^{n-1}}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a)}\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right] - \frac{\eta_1}{(1-\eta_1)}\sum_i^{-a}\mathcal{U}^i(u_s^i, \tau^i)$$

$$\tag{49}$$

Substituting Eq.49 into Eq.48, we have

$$
\begin{aligned}
\mathcal{U}^a(u_s^a, \tau^a) =& (1 - \eta_{1,w})(1-\alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \frac{1 - \eta_{1,w}}{1 - \eta_1}\sum_k^{m^{n-1}}\left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a)}\sum_i^{-a}\mathcal{U}^i(u_k^i, \tau^i)\right] \\
&+ \frac{\eta_1 - \eta_{1,w}}{1 - \eta_1}\sum_i^{-a}\mathcal{U}^i(u_s^i, \tau^i) + \eta_{1,w}\mathcal{Q}_{its}(s, \boldsymbol{u}_s)
\end{aligned}
\tag{50}
$$

Notice that

$$\sum_{a=1}^{n} \sum_{i}^{-a} \mathcal{U}^i(u_s^i, \tau^i) = (n-1) \sum_{a=1}^{n} \mathcal{U}^i(u_s^i, \tau^i) = (n-1)Q(\boldsymbol{u}_s, \boldsymbol{\tau}) \tag{51}$$

Therefore,

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) = \sum_{a=1}^{n} \mathcal{U}^a(u_s^a, \tau^a) = n(1-\eta_{1,w})(1-\alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + (n-1)\frac{\eta_1 - \eta_{1,w}}{1-\eta_1}Q(\boldsymbol{u}_s, \boldsymbol{\tau})$$
$$+ n\eta_{1,w}\mathcal{Q}_{its}(s, \boldsymbol{u}_s) - \frac{1-\eta_{1,w}}{1-\eta_1} \sum_{a=1}^{n} \sum_{k}^{m^{n-1}} \left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a)} \sum_{i}^{-a} \mathcal{U}^i(u_k^i, \tau^i) \right] \tag{52}$$

According to Eq.31, we have

$$\sum_{a=1}^{n} \sum_{k}^{m^{n-1}} \left[\frac{p(\acute{u}^a, u_k^{-a})}{p(\acute{u}^a)} \sum_{i}^{-a} \mathcal{U}^i(u_k^i, \tau^i) \right] = n(1-\eta_2)(1-\alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\eta_2 \mathcal{Q}(s, \acute{\boldsymbol{u}}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \tag{53}$$

Substituting Eq.53 into Eq.52, we have

$$\left[1 - (n-1)\frac{\eta_1 - \eta_{1,w}}{1-\eta_1}\right]Q(\boldsymbol{u}_s, \boldsymbol{\tau}) = n(1-\eta_{1,w})(1-\alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\eta_{1,w}\mathcal{Q}_{its}(s, \boldsymbol{u}_s)$$
$$- \frac{1-\eta_{1,w}}{1-\eta_1}\left[n(1-\eta_2)(1-\alpha) - 1\right]Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - n\eta_2 \frac{1-\eta_{1,w}}{1-\eta_1}\mathcal{Q}(s, \acute{\boldsymbol{u}}) \tag{54}$$

where $\eta_2 = (1 - \epsilon + \frac{\epsilon}{m})^{n-1}$. Eq.54 can be further simplified

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) = \frac{n(1-\alpha)(\eta_2 - \eta_1) + 1}{1 + n(w-1)\eta_1}Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\frac{w(1+e_Q)\eta_1 - \eta_2}{1 + n(w-1)\eta_1}\mathcal{Q}(s, \acute{\boldsymbol{u}}) \tag{55}$$

Finally, we have

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) = n\frac{(1-\alpha)(\eta_2 - \eta_1) - (w-1)\eta_1}{1 + n(w-1)\eta_1}Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\frac{w(1+e_Q)\eta_1 - \eta_2}{1 + n(w-1)\eta_1}\mathcal{Q}(s, \acute{\boldsymbol{u}}) \tag{56}$$

When $\boldsymbol{w} = 1$, Eq.56 degenerates to Eq.32. For monotonic value decomposition, Eq.56 also holds since $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ and $Q(\boldsymbol{u}_s, \boldsymbol{\tau})$ do not change.

Since $\acute{\boldsymbol{u}}$ is a non-optimal action, according to *Conidtion* 2 (Eq.5, Section 3.2), to ensure this non-optimal point is not an STN, let $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) > Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ and assume $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \approx \mathcal{Q}(s, \acute{\boldsymbol{u}})$ (this assumption is quite accurate under ITS, as verified in Appendix F.2), we have

$$w > \frac{\alpha(\eta_2 - \eta_1)}{e_{Q0}\eta_1} = w_0 \tag{57}$$

Therefore, the non-optimal STNs can be eliminated by applying a large enough weight on the superior actions under ITS. For the non-hardest exploration cases, η_1 in Eq.57 is replaced with η_1'. Since $\frac{\alpha(\eta_2 - \eta_1)}{e_{Q0}\eta_1} > \frac{\alpha(\eta_2 - \eta_1')}{e_{Q0}\eta_1'}$, we only need to consider the hardest exploration case.

F.2. Verification

We carry out experiments in matrix games to evaluate the effect of weights on the superior actions under ITS. The pay-off matrix is defined as

$$\mathcal{Q}(s, \boldsymbol{u}) = \begin{cases} 6(1+e_Q) & \boldsymbol{u} = \{0, 0\} \\ 6 & \boldsymbol{u} = \{m, m\} \\ random(-20, 6) & others \end{cases} \tag{58}$$

where m is the size of individual action space. The greedy action is fixed to $\acute{\boldsymbol{u}} = \{m, m\}$. Therefore, $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) = 6$. An mlp shared by all agents is adopted as the agent network. 1000 iterations (100 episodes per iteration) of training over 5 seeds are executed on each set of parameters, where $\alpha = 0.1$, $\epsilon = 0.2$ and $e_Q = 1/3$.

Table 4. Comparison between test and calculation (shown in parentheses) $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ on n-agent matrix games when $w = w_0$.

m^n	3^2	5^2	10^2	3^3	3^4
w_0 (Eq.57)	3.60	6.00	12.00	50.32	**659.50**
$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ (Eq.56)	0.01 $\pm$0.06 (0)	0.02 $\pm$0.16 (0)	0.22 $\pm$0.13 (0)	-0.02 $\pm$0.30 (0)	-0.48 $\pm$0.75 (0)
Test $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$	5.95 $\pm$0.02	5.97$\pm$0.02	5.98 $\pm$0.01	5.90 $\pm$0.06	5.93 $\pm$0.03

The experimental results are shown in Tab.4. Firstly, the error of $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ between test and calculation is very small. Secondly, the joint Q value of the greedy action approximates its true Q value, i.e., $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \approx \mathcal{Q}(s, \acute{\boldsymbol{u}}) = 6$) under ITS. Thirdly, the lower bound of w grows *exponentially* as the number of agent grows, which introduces instability in $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$.

G. STNs under ITS with Superior Experience Replay

Given the greedy action $\{\acute{u}^1, \cdots, \acute{u}^n\} = \acute{\boldsymbol{u}}$ and a superior action $\{u_s^1, \cdots, u_s^n\} = \boldsymbol{u}_s \neq \acute{\boldsymbol{u}}$, we have $\mathcal{Q}(s, \boldsymbol{u}_s) > \mathcal{Q}(s, \acute{\boldsymbol{u}})$. Here we regard all actions except $\boldsymbol{u}_s$ and $\acute{\boldsymbol{u}}$ as inferior actions. Discussions under multiple superior actions is provided in Appendix H.1. We first consider the hardest exploration case. For simplicity, assuming $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) > 0$, where $\mathcal{Q}_{its}(s, \boldsymbol{u}) = (1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ for inferior actions. By applying a weight w_{ser} on the loss of superior actions from the superior buffer, the utility function of individual action $u_s^a (a \in [1, n])$ consists of two parts

$$\mathcal{U}^a(u_s^a, \tau^a) = (1 - \eta_{1,ser}) \left[(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - \sum_k^{m^{n-1}-1} \left[\frac{p(u_s^a, u_k^{-a})}{p(u_s^a) - p(\boldsymbol{u}_s)} \sum_i^{-a} \mathcal{U}^i(u_k^i, \tau^i) \right] \right]$$
$$+ \eta_{1,ser} \left[\mathcal{Q}_{its}(s, \boldsymbol{u}_s) - \sum_i^{-a} \mathcal{U}^i(u_s^i, \tau^i) \right]$$
(59)

where $\eta_{1,ser} = \frac{w_{ser} + \eta_1 \eta_s}{\eta_s + w_{ser}}$. $\eta_1 = (\frac{\epsilon}{m})^{n-1}$, and η_s is the probability of state s. Please refer to Appendix D.2 for more details about the notations. Following the derivation provided in Appendix F.1, we have

$$\left[1 - (n-1)\frac{\eta_1 - \eta_{1,ser}}{1 - \eta_1} \right] Q(\boldsymbol{u}_s, \boldsymbol{\tau}) = n(1 - \eta_{1,ser})(1 - \alpha)Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\eta_{1,ser}\mathcal{Q}_{its}(s, \boldsymbol{u}_s)$$
$$- \frac{1 - \eta_{1,ser}}{1 - \eta_1} \left[n(1 - \eta_2)(1 - \alpha) - 1 \right] Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) - n\eta_2 \frac{1 - \eta_{1,ser}}{1 - \eta_1} \mathcal{Q}(s, \acute{\boldsymbol{u}})$$
(60)

Eq.60 can be further simplified

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) = \eta_s \frac{n(1 - \alpha)(\eta_2 - \eta_1) + 1}{\eta_s + n w_{ser}} Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\frac{(w_{ser} + \eta_1 \eta_s)(1 + e_Q) - \eta_2 \eta_s}{\eta_s + n w_{ser}} \mathcal{Q}(s, \acute{\boldsymbol{u}})$$
(61)

where $\eta_2 = (1 - \epsilon + \frac{\epsilon}{m})^{n-1}$. Therefore, we have

$$Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) = n\frac{(1 - \alpha)(\eta_2 - \eta_1)\eta_s - w_{ser}}{\eta_s + n w_{ser}} Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) + n\frac{(w_{ser} + \eta_1 \eta_s)(1 + e_Q) - \eta_2 \eta_s}{\eta_s + n w_{ser}} \mathcal{Q}(s, \acute{\boldsymbol{u}})$$
(62)

When $w = 0$, Eq.62 degenerates to Eq.32. For monotonic value decomposition, Eq.62 also holds since $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ and $Q(\boldsymbol{u}_s, \boldsymbol{\tau})$ do not change.

Since $\acute{\boldsymbol{u}}$ is a non-optimal action, according to *Conidtion* 2 (Eq.5, Section 3.2), to ensure this non-optimal point is not an STN, let $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) > Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ and assume $Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau}) \approx \mathcal{Q}(s, \acute{\boldsymbol{u}})$ (this assumption is quite accurate under ITS, as verified in Appendix F.2), we have

$$w_{ser} > \frac{\alpha}{e_{Q0}}(\eta_2 - \eta_1)\eta_s - \eta_1 \eta_s$$
(63)

According to Eq.63, SER can eliminate the non-optimal STNs by selecting a large enough w for the loss of superior actions from the superior buffer. For the non-hardest exploration cases, η_1 in Eq.57 is replaced with η_1'. Since $\eta_1' > \eta_1$, we only need to consider the hardest exploration case.

H. Discussions

H.1. GVR under Multiple Superior Actions

In previous derivations, we regard all actions except $\boldsymbol{u}_s$ and $\acute{\boldsymbol{u}}$ as inferior actions, i.e., we only consider the cases with no more than one superior action. For situations with multiple superior actions $\{\boldsymbol{u}_{s,1}, \cdots, \boldsymbol{u}_{s,p}\}$, two examples where the condition in Eq.47 fails to eliminate the non-optimal STNs are given in Fig.11.
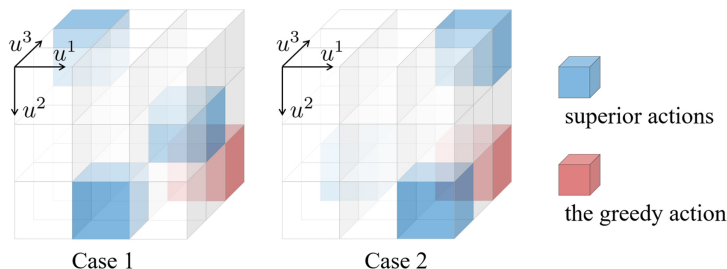

Figure 11. Failure cases for condition in Eq.47 under multiple superior actions, where the number of agents and the size of individual action space are both 3. The superior actions ($\{\boldsymbol{u}_{s,1}, \cdots, \boldsymbol{u}_{s,p}\}$) and greedy action $\acute{\boldsymbol{u}}$) are denoted by blocks of blue and red respectively.

In both examples, $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$ is smaller than any cases with only one superior action. Eq.47 is an insufficient condition for $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) > Q(\acute{\boldsymbol{u}}, \boldsymbol{\tau})$. Since most of the superior actions in training batch come from superior buffer, to avoid the cases with multiple superior actions, we set the superior batch size to 1.

H.2. Trade-off between Optimality and Stability

The optimal consistency (or TGM principle) requires access to the true Q values, which are usually obtained by estimation. In such cases, excessive pursuit for optimality may decrease the stability. An example is shown in Fig.12.
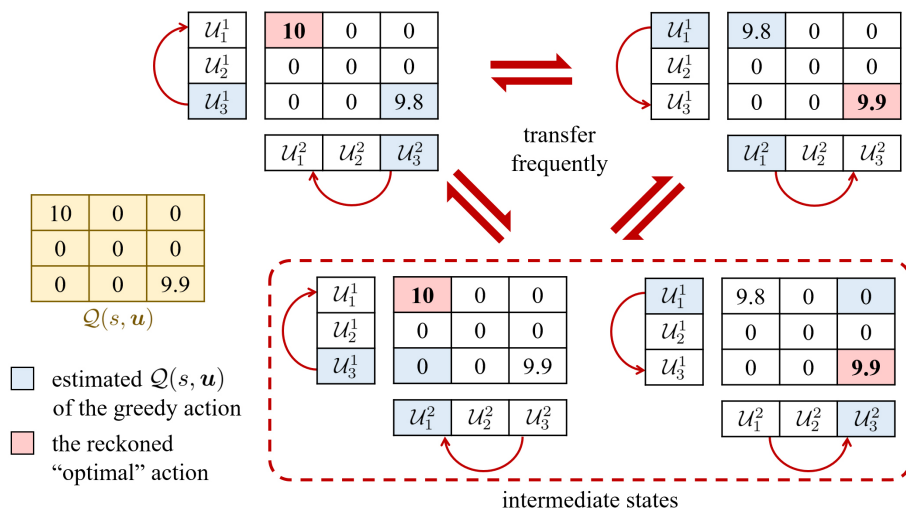

Figure 12. An example of instability caused by excessive pursuit for optimality. Due to the estimation error of the true Q values, actions with large true Q values may be mistaken for the optimal action. As a result, the joint policy transfers frequently between the reckoned "optimal" actions. Worse, the out-of-step updates of individual polices put the joint policy at poor intermediate states.

I. The Working Principle of GVR

The working principle of GVR is shown in Fig.13. Please refer to Appendix G for more details about the notations.
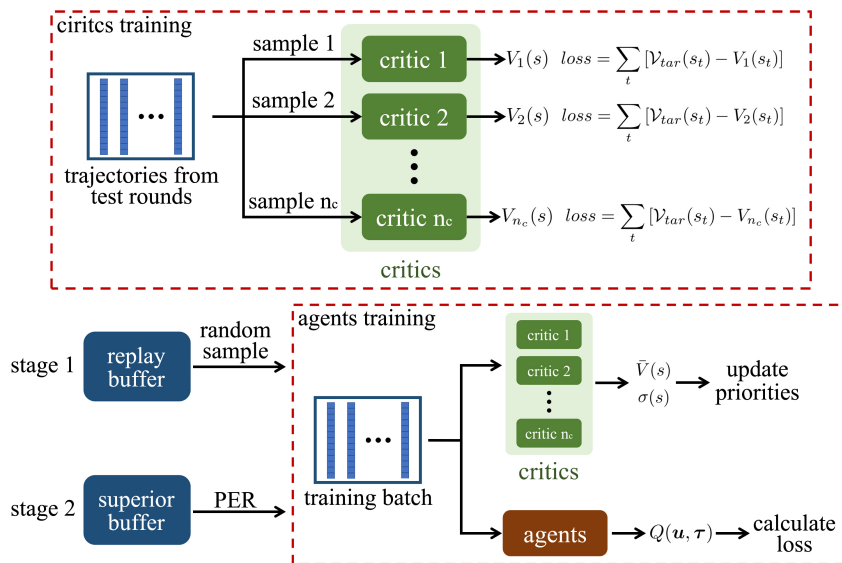
Figure 13. The working principle of GVR. The critics are trained on the trajectories sampled from test rounds. All critics share the same target $\mathcal{V}_{gvr}(s)$ (Eq.11). The agents training consists of two stages. In stage 1, the training batch is randomly sampled from the replay buffer, where the loss function is $loss = \sum_t \left[\mathcal{Q}_{its}(s_t, \boldsymbol{u}_t) - Q(\boldsymbol{u}_t, \boldsymbol{\tau}_t) \right]$. In stage 2, the training batch is sampled from the superior buffer with prioritized experience replay (PER), where the loss function is $loss = \sum_t w(s_t) \mathbb{I}_{sup} \left[\mathcal{Q}_{its}(s_t, \boldsymbol{u}_t) - Q(\boldsymbol{u}_t, \boldsymbol{\tau}_t) \right]$. $\mathbb{I}_{sup}(s_t, \boldsymbol{u}_t)$ is a indicator for the superior action, i.e., $\mathbb{I}_{sup}(s_t, \boldsymbol{u}_t) = 1$, $s.t. \sum_{t=t_0}^{T} \gamma^{t-t_0} r(s_t, \boldsymbol{u}_t) > \bar{V}(s_t) + 3\sigma(s_t)$. $w_{ser}(s_t) = \frac{\alpha}{e_{Q0}}(\eta_2 - \eta_1) - \eta_1$ for hardest exploration cases and $w_{ser}(s_t) = \frac{\alpha}{e_{Q0}}(\eta_2 - \eta_1') - \eta_1'$ for non-hardest exploration cases, where $e_{Q0} = \frac{3\sigma(s)}{V(s)}$. At the end of both stages, the trajectories in the training batch are stored into the superior buffer after the update of their priorities.

J. Experimental Settings and Additional Experiments

J.1. Experimental Settings

In experiments on one-step matrix games, since the episode length is 1, an mlp shared by all agents is adopted as the agent network. Besides, $\eta_s = 1$ since there is only 1 state. No data buffer is used in the verification of calculation results, e.g., $Q(\boldsymbol{u}_s, \boldsymbol{\tau}) - Q(\hat{\boldsymbol{u}}, \boldsymbol{\tau})$ (Fig.2(a), Section 5.1) and $Q(\boldsymbol{u}, \boldsymbol{\tau})$ (Tab.3, Appendix C.2). Otherwise, a replay buffer of length 1000 of is applied matrix for all algorithms, e.g., Fig.2(b) (Section 5.1) and Fig.3 (Section 5.1). For WQMIX, $\alpha = 0.5$. For GVR, $\alpha = 0.2$ and the length of superior buffer is 3. All experiments are carried out over 5 seeds.

In experiments on predator-prey and SMAC, we adopt the default settings for VDN, QMIX, QPLEX and WQMIX. The length of replay buffer is 5000 and the batch size is 32. For WQMIX, $\alpha = 0.5$. For GVR, $\alpha = 0.2$ and the length of superior buffer is 300. According to Appendix H.1, we set the size of superior batch to 1. The probability of a state η_s is unknown, a sufficient condition of Eq.63 is $\eta_s = 1$. All experiments are carried out over 5 seeds.

The game version of StarCraft II is 69232. Each algorithm is trained for 2e6 steps in MMM2, 2c_vs_64_zg and 6h_vs_8z, with ϵ damping from 1 to 0.05 in the first 5e4 steps. Besides, in 6h_vs_8z, 3s5z_vs_3s6z and corridor, each algorithm is trained for 5e6 steps, with ϵ damping from 1 to 0.05 in the first 1e6 steps.

J.2. Ablation Studies

We conduct ablation studies to investigate the improvements of GVR. We first evaluate the effect of inferior target shaping (ITS) and superior experience replay (SER) on predator-prey. We apply a constant weight $w = 3$ to the loss of action $\boldsymbol{u}_t$ for ITS when $Q(\boldsymbol{u}_t, \boldsymbol{\tau}_t) < r(s_t, \boldsymbol{u}_t) + \gamma Q(\boldsymbol{u}_{t+1}, \boldsymbol{\tau}_{t+1})$. As shown in Fig.14(a), in task with punishment -2, both ITS and SER help to overcome the relative overgeneralization (RO). In task with punishment -5, SER alone is unable to overcome the RO since the STNs depend on the true Q values of inferior actions. Although ITS remove the dependence, it can not ensure the optimal consistency without SER.

We also investigate the effect of the parameter α. As shown in Fig.14(b), a too small or too large α leads to poor performance.
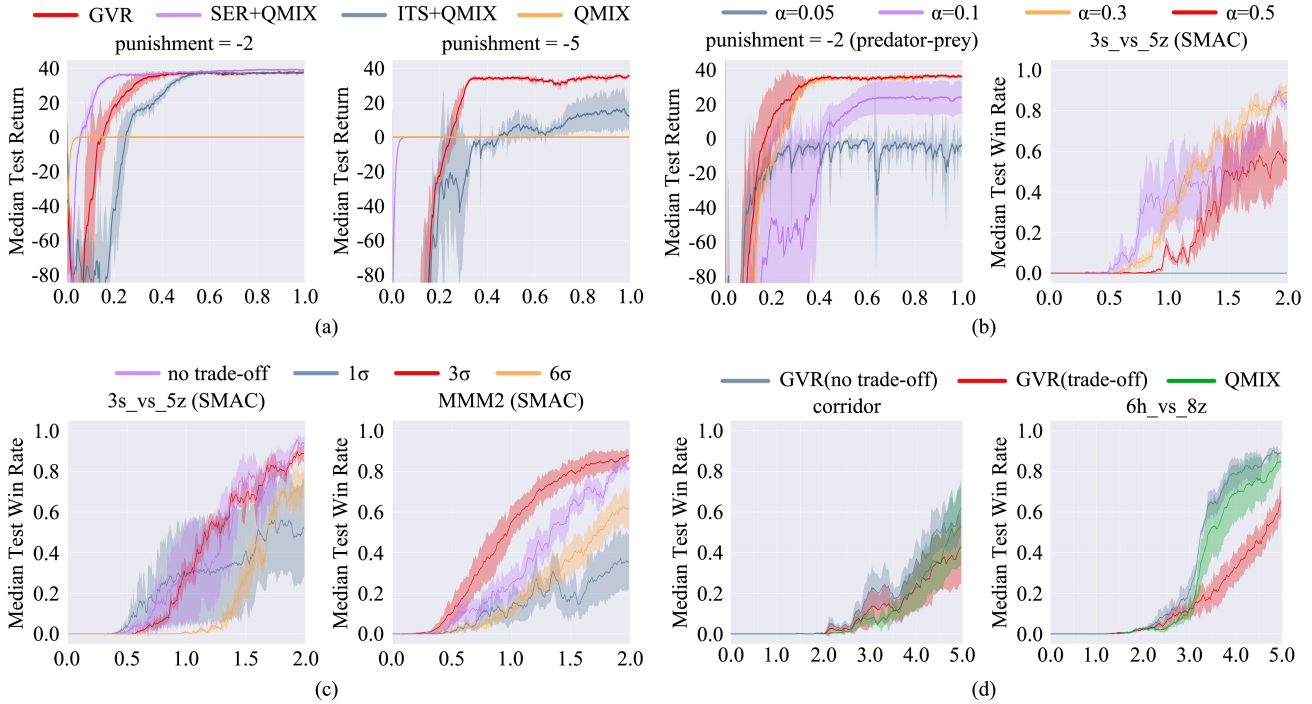
Figure 14. Ablation studies. The x-axes are training time-steps (million). (a) Effect of ITS and SER. Investigation on parameter (b) α and (c) e_{Q0}. (d) GVR with trade-off (3σ) vs GVR without trade-off in hard exploration tasks.

Since α defines the gap of the joint Q values' targets between the inferior and greedy actions, a too small α causes confusion between the greedy and the inferior actions. Meanwhile, a too large α prevents the update from a greedy action to a superior action.

To find a suitable hyper-parameter for the trade-off between stability and optimality, we compare the performance of GVR under different $e_{Q0)}$. The experimental results are shown in Fig.14(c), where $k\sigma$ denotes $e_{Q0} = \frac{k\sigma(s)}{\bar{V}(s)}$ ($k \in \{1, 3, 6\}$). The algorithm attaches more importance to stability as k increases. In the experiments of $no\ trade-off$, we do not use the ensemble critics, where the joint Q value function is applied to identify inferior and superior actions. In this case, an action u_t is classified into superior action when $u_t \neq \acute{u}$ and $r(s_t, u_t) + \gamma Q(u_{t+1}, \tau_{t+1}) > Q(\acute{u}_t, \tau_t)$. As shown in Fig.14(c), a proper e_{Q0} is helpful for the balance of stability and optimality. However, the trade-off do not always helps. As shown in Fig.14(d), in hard exploration tasks, GVR with trade-off do not perform better than GVR without trade-off.
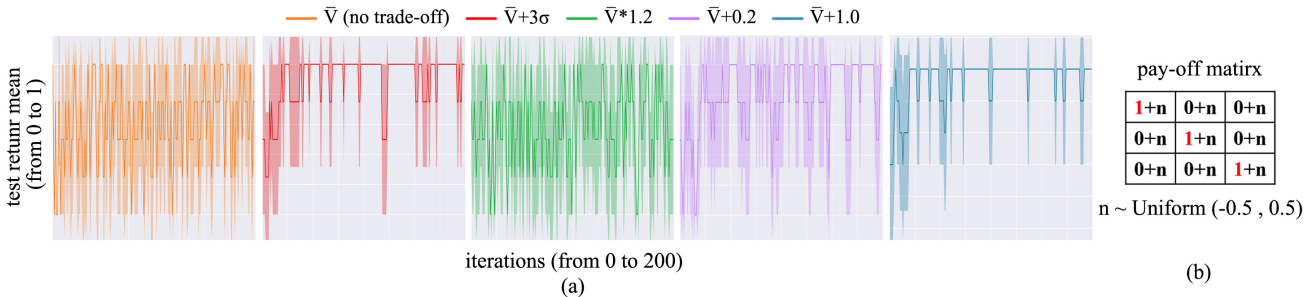

Figure 15. Investigations of threshold functions in a two-agent matrix game. The optimal node changes in training due to a noise applied to the rewards. As a result, the greedy action may jump frequently between the reckoned optimal nodes and poor intermediates.

In the experiments of GVR with trade-off, we find the critics often conflict with the joint Q value function, e.g., $r(s_t, \boldsymbol{u}_t) + \gamma Q(\boldsymbol{u}_{t+1}, \boldsymbol{\tau}_{t+1}) < Q(\boldsymbol{u}_t, \boldsymbol{\tau}_t)$ happens for a superior action $\boldsymbol{u}_t$. The ensemble critics and the joint Q value function are both evaluations of the joint policy. We mix these two evaluations to the achieve the trade-off in GVR. The conflict between two evaluations may cause performance deterioration. As a result, GVR with trade-off performs worse than that without trade-off in some tasks. The latter adopts the joint Q value function as the unique evaluation.

For the adaptive trade-off, we investigate some other threshold functions, such as $\bar{V} + C$ and $(1 + C) * \bar{V}$, where $C > 0$ is a predefined parameter. However, the scalability of these threshold functions is poor because (1) the suitable C varies as the reward scale; (2) the estimation error of the critics decreases as training, where C requires attenuation accordingly. Empirical results are shown in Fig.1. A well-designed threshold ($\bar{V} + 1.0$) may perform better but requires prior knowledge about the value's gap between the optimal and sub-optimal actions.

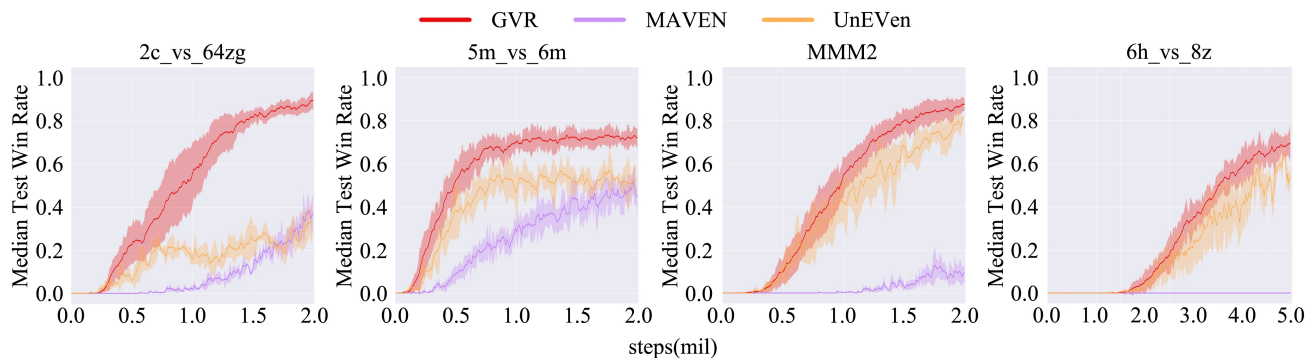### J.3. Comparison With Joint Exploration Methods

Figure 16. Comparison between GVR, MAVEN and UneVEn.

We compare our method with joint exploration methods on SMAC. The experimental results are shown in Fig.16.