SHAQ: Incorporating Shapley Value Theory into Multi-Agent Q-Learning

Jianhong Wang
Imperial College London, UK
jianhong.wang16@imperial.ac.uk

Yuan Zhang
University of Freiburg, Germany
yzhang@cs.uni-freiburg.de

Yunjie Gu*
University of Bath, UK
yg934@bath.ac.uk

Tae-Kyun Kim
KAIST, South Korea
kimtaekyun@kaist.ac.kr

Abstract

Value factorisation is a useful technique for multi-agent reinforcement learning (MARL) in global reward game, however, its underlying mechanism is not yet fully understood. This paper studies a theoretical framework for value factorisation with interpretability via Shapley value theory. We generalise Shapley value to Markov convex game called *Markov Shapley value* (MSV) and apply it as a value factorisation method in global reward game, which is obtained by the equivalence between the two games. Based on the properties of MSV, we derive *Shapley-Bellman optimality equation* (SBOE) to evaluate the optimal MSV, which corresponds to an optimal joint deterministic policy. Furthermore, we propose *Shapley-Bellman operator* (SBO) that is proved to solve SBOE. With a stochastic approximation and some transformations, a new MARL algorithm called *Shapley Q-learning* (SHAQ) is established, the implementation of which is guided by the theoretical results of SBO and MSV. We also discuss the relationship between SHAQ and relevant value factorisation methods. In the experiments, SHAQ exhibits not only superior performances on all tasks but also the interpretability that agrees with the theoretical analysis. The implementation of this paper is placed on `https://github.com/hsvgbkhgbv/shapley-q-learning`.

1 Introduction

Cooperative games are a critical research area in multi-agent reinforcement learning (MARL). Many real-life tasks can be modeled as cooperative games, e.g. the coordination of autonomous vehicles [1], autonomous distributed logistics [2] and distributed voltage control in power networks [3]. In this paper, we consider global reward game (a.k.a. team reward game), an important subclass of cooperative games, wherein agents aim to jointly maximize cumulative global rewards over time. There are two categories of methods to solve this problem: (i) each agent identically maximizes cumulative global rewards, i.e. learning with a shared value function [4–6]; and (ii) each agent individually maximizes distributed values, i.e. learning with (implicit) credit assignments (e.g. marginal contribution and value factorisation) [7–11].

By the view of non-cooperative game theory, global reward game are equivalent to Markov game [12] with global reward (a.k.a. team reward). Its aim is to learn a stationary joint policy to reach a Markov equilibrium so that no agent tends to unilaterally change its policy to maximize cumulative global rewards. Standing by this view, learning with value factorisation cannot be directly explained [13]. In

*Correspondence to Yunjie Gu who is also an honorary lecturer at Imperial College London.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

this paper, to clearly interpret the value factorisation, we take the perspective of cooperative game theory [14], wherein agents are partitioned into coalitions and a payoff distribution scheme is found to distribute optimal values to coalitions. The corresponding solution is called Markov core, whereby no agent has an incentive to deviate. When all agents are partitioned into one coalition (called grand coalition), the payoff distribution scheme naturally plays the role of value factorisation.

Wang et al. [13] extended convex game (i.e. a game model in cooperative game theory) [14] to dynamic scenarios, which we name as Markov convex game in this paper. We construct the analytic form of Shapley value for Markov convex game, and prove that it reaches the Markov core under the grand coalition, named as Markov Shapley value. The optimal Markov Shapley value implies not only the optimal global value but also that no agent has incentives to deviate from the grand coalition. Additionally, Markov Shapley value enjoys the following properties: (i) identifiability of dummy agents; (ii) efficiency; (iii) reflecting the contribution; and (iv) symmetry. These properties aid the interpretation and validity of value factorisation in the global reward game, and such transparency and reliability are critical to industrial applications [3].

Based on the efficiency property, we derive Shapley-Bellman optimality equation that is an extension of Bellman optimality equation [15, 16]. Moreover, we propose Shapley-Bellman operator and prove its convergence to the Shapley-Bellman optimality equation and its optimal joint deterministic policy. With a stochastic approximation of Shapley-Bellman operator and some transformations, we derive an algorithm called Shapley Q-learning (SHAQ). SHAQ learns to approximate the optimal Markov Shapley Q-value (an equivalent form of the optimal Markov Shapley value). Moreover, we enable SHAQ decentralised in order to fit the decentralised execution framework and this decentralisation still remains the convergence condition of Shapley-Bellman operator.

The proposed method, SHAQ, is evaluated on two global reward games such as Predator-Prey [17] and multi-agent StarCraft benchmark tasks [18]. In the experiments, SHAQ shows not only generally good performances on solving all tasks but also the interpretability that is deficient in the state-of-the-art baselines.

2 Markov Convex Game

We now formally define Markov convex game (MCG) that can be described as a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, T, \Lambda, \pi, R_t, \gamma \rangle$. $\mathcal{N}$ is the set of all agents. $\mathcal{S}$ is the set of states and $\mathcal{A} = \times_{i \in \mathcal{N}} \mathcal{A}_i$ is the joint action set of all agents wherein $\mathcal{A}_i$ is each agent's action set. $T(\mathbf{s}, \mathbf{a}, \mathbf{s}') = Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ is defined as the transition probability between the successive states. $\mathcal{CS} = \{\mathcal{C}_1, ..., \mathcal{C}_n\}$ is a *coalition structure*, where $\mathcal{C}_i \subseteq \mathcal{N}$ called a *coalition* is a subset of all agents. Λ is a collection of coalition structures. $\varnothing$ and $\mathcal{N}$ are two special cases of coalitions i.e. the *empty coalition* and the *grand coalition* respectively. Conventionally, it is assumed that $\mathcal{C}_m \bigcap \mathcal{C}_k = \varnothing, \forall \mathcal{C}_m, \mathcal{C}_k \subseteq \mathcal{N}$. $\pi = \times_{i \in \mathcal{N}} \pi_i$ is the joint policy of all agents. For any coalition $\mathcal{C}$, it is equipped with a *coalition policy* $\pi_{\mathcal{C}}(\mathbf{a}_{\mathcal{C}}|\mathbf{s}) = \times_{i \in \mathcal{C}} \pi_i(a_i|\mathbf{s})$ defined over the *coalition action set* $\mathcal{A}_{\mathcal{C}} = \times_{i \in \mathcal{C}} \mathcal{A}_i$. Therefore, π can be seen as the *grand coalition policy*. $R_t : \mathcal{S} \times \mathcal{A}_{\mathcal{C}} \to [0, \infty)$ (i.e., a characteristic function) is the *coalition reward* at time step t. Accordingly, $R_t(\mathbf{s}, \mathbf{a})$ is the *grand coalition reward* (i.e., equivalent to the global reward) at time step t that is written as $R(\mathbf{s}, \mathbf{a})$ or R for conciseness in the rest of paper. $\gamma \in (0, 1)$ is the discounted factor. The infinite long-term discounted cumulative coalition rewards is defined as $V^{\pi_{\mathcal{C}}}(\mathbf{s}) = \mathbb{E}_{\pi_{\mathcal{C}}}\big[\sum_{t=1}^{\infty} \gamma^{t-1} R_t(\mathbf{s}, \mathbf{a}_{\mathcal{C}}) \mid \mathbf{S}_t = \mathbf{s}\big] \in [0, \infty)$, called a *coalition value*. Moreover, the empty coalition value $V^{\pi_{\varnothing}}(\mathbf{s}) = 0$ and $V^{\pi}(\mathbf{s})$ denotes the grand coalition value (i.e. also called the global value since the equivalence proof from [13]). The solution of MCG is to find a tuple $\big\langle \mathcal{CS}, (\max_{\pi_i} x_i(\mathbf{s}))_{i \in \mathcal{N}} \big\rangle$, where $(\max_{\pi_i} x_i(\mathbf{s}))_{i \in \mathcal{N}}$ indicates the *payoff distributions* (i.e. credit assignments) under the optimal joint policy given a coalition structure. Under the assumption $\mathcal{C}_m \bigcap \mathcal{C}_k = \varnothing, \forall \mathcal{C}_m, \mathcal{C}_k \subseteq \mathcal{N}$, the condition for MCG is as follows:

$$\max_{\pi_{\mathcal{C}_\cup}} V^{\pi_{\mathcal{C}_\cup}}(\mathbf{s}) \geq \max_{\pi_{\mathcal{C}_m}} V^{\pi_{\mathcal{C}_m}}(\mathbf{s}) + \max_{\pi_{\mathcal{C}_k}} V^{\pi_{\mathcal{C}_k}}(\mathbf{s}), \quad \forall \mathcal{C}_m, \mathcal{C}_k \subseteq \mathcal{N}, \mathcal{C}_\cup = \mathcal{C}_m \cup \mathcal{C}_k. \tag{1}$$

In MCG with the grand coalition i.e., $\mathcal{CS} = \{\mathcal{N}\}$, *Markov core*, a solution concept describing stability, is defined as a set of payoff distribution schemes by which no agent has incentives to deviate from the grand coalition to gain more profits. Mathematically, Markov core can be expressed as:

$$\texttt{MarkovCore} = \Big\{ \big(\max_{\pi_i} x_i(\mathbf{s}) \big)_{i \in \mathcal{N}} \Big| \max_{\pi_{\mathcal{C}}} x(\mathbf{s}|\mathcal{C}) \geq \max_{\pi_{\mathcal{C}}} V^{\pi_{\mathcal{C}}}(\mathbf{s}), \forall \mathcal{C} \subseteq \mathcal{N}, \mathbf{s} \in \mathcal{S} \Big\}, \tag{2}$$

where $\max_{\pi_\mathcal{C}} x(\mathbf{s}|\mathcal{C}) = \sum_{i\in\mathcal{C}} \max_{\pi_i} x_i(\mathbf{s})$. It aims to find a payoff distribution scheme $(x_i(\mathbf{s}))_{i\in\mathcal{N}}$ that can finally converge to Markov core under the optimal joint policy.

To assist the application on Q-learning, we similarly define *coalition Q-value* as $Q^{\pi_\mathcal{C}}(\mathbf{s}, \mathbf{a}_\mathcal{C}) \in [0, +\infty)$ for all coalitions $\mathcal{C} \subset \mathcal{N}$. Following the above convention, the grand coalition Q-value (or the global Q-value) can be written as $Q^\pi(\mathbf{s}, \mathbf{a})$. Moreover, the optimal coalition Q-value of $\mathcal{C}$ w.r.t. the optimal joint policy of $\mathcal{D} \subseteq \mathcal{C}$ (i.e., $\pi^*_\mathcal{D}$) and the suboptimal joint policy of $\mathcal{C}\backslash\mathcal{D}$ (i.e., $\pi_{\mathcal{C}\backslash\mathcal{D}}$) is defined as $Q^{\pi^*_\mathcal{D}}(\mathbf{s}, \mathbf{a}_\mathcal{C})$. Therefore, the optimal coalition Q-value of $\mathcal{C}$ w.r.t. the optimal joint policy of $\mathcal{C}$ is defined as $Q^{\pi^*_\mathcal{C}}(\mathbf{s}, \mathbf{a}_\mathcal{C})$. Accordingly, the optimal global coalition Q-value w.r.t. the optimal joint policy of the grand coalition is denoted as $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$.

3 Markov Shapley Value

By the view of cooperative game theory, the grand coalition is progressively formed by a permutation of agents. Accordingly, marginal contribution is an implementation of the credit reflecting an agent's contribution. The formal definition is shown in Definition 1.

Definition 1. *In Markov convex game, with a permutation of agents $\langle j_1, j_2, ..., j_{|\mathcal{N}|}\rangle$, $\forall j_n \in \mathcal{N}$ forming the grand coalition $\mathcal{N}$, where $n \in \{1, ..., |\mathcal{N}|\}$, $j_a \neq j_b$ if $a \neq b$, the marginal contribution of an agent i is defined as the following equation such that*

$$\Phi_i(\mathbf{s}|\mathcal{C}_i) = \max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i\cup\{i\}}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s}), \tag{3}$$

where $\mathcal{C}_i = \{j_1, ..., j_{n-1}\}$ for $j_n = i$ is an arbitrary intermediate coalition where agent i would join during the process of grand coalition formation.

Proposition 1. *Agent i's action marginal contribution can be derived as follows:*

$$\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}). \tag{4}$$

As Proposition 1 shows, an agent's action marginal contribution (analogous to Q-value) can be derived according to Eq.4. It is usually more useful for solving MARL problems.

It is apparent that marginal contribution only considers one permutation to form the grand coalition. By the viewpoint from Shapley [19], the fairness is achieved through considering how much the agent i increases the optimal values (i.e. marginal contributions) of the coalitions in all possible permutations when it joins in, i.e., $\max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i\cup\{i\}}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s}), \forall \mathcal{C}_i \subseteq \mathcal{N}\backslash\{i\}$. Therefore, we construct Shapley value under Markov dynamics based on the marginal contributions shown in Definition 2, named as *Markov Shapley value* (MSV).

Definition 2. *Markov Shapley value is represented as*

$$V^\phi_i(\mathbf{s}) = \sum_{\mathcal{C}_i \subseteq \mathcal{N}\backslash\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \Phi_i(\mathbf{s}|\mathcal{C}_i). \tag{5}$$

With the deterministic policy, Markov Shapley value can be equivalently represented as

$$Q^\phi_i(\mathbf{s}, a_i) = \sum_{\mathcal{C}_i \subseteq \mathcal{N}\backslash\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \Phi_i(\mathbf{s}, a_i|\mathcal{C}_i). \tag{6}$$

where $\Phi_i(\mathbf{s}|\mathcal{C}_i)$ is defined in Eq.3 and $\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i)$ is defined in Eq.4.

For convenience, we name Eq.6 as *Markov Shapley Q-value* (MSQ). Briefly, MSV calculates the weighted average of marginal contributions. Since a coalition may repeatedly appear among all permutations (i.e. $|\mathcal{N}|!$ permutations), the ratio between the occurrence frequency $|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!$ and the total frequency $|\mathcal{N}|!$ is used as a weight to describe the importance of the corresponding marginal contribution. Besides, the sum of all weights is equal to 1, so each weight can be interpreted as a probability distribution. Consequently, MSV can be seen as the expectation of marginal contributions, denoted as $\mathbb{E}_{\mathcal{C}_i\sim Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})}[\Phi_i(\mathbf{s}|\mathcal{C}_i)]$. Note that $Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ is a bell-shaped probability distribution. By the above relationship, Remark 1 is directly obtained.

Remark 1. *Uniformly sampling different permutations is equivalent to directly sampling from $Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$, since the coalition generation is from the permutation to form the grand coalition.*

Proposition 2. *Markov Shapley value possesses properties as follows: (i) identifiability of dummy agents:* $V_i^{\phi}(\mathbf{s}) = 0$*; (ii) efficiency:* $\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i\in\mathcal{N}} \max_{\pi_i} V_i^{\phi}(\mathbf{s})$*; (iii) reflecting the contribution; and (iv) symmetry.*

Proposition 2 shows four properties of MSV. The most important property is Property (ii) that aids the formulation of Shapley-Bellman optimality equation. Property (iii) shows that MSV is a fundamental index to quantitatively describe each agent's contribution. Property (i) and (iii) play important roles in interpretation for value factorisation (or credit assignment). Property (iv) indicates that if two agents are symmetric, then their optimal MSVs should be equal, *but the reverse does not necessarily hold*. All these properties that define the fairness are inherited from the original Shapley value [19].

4 Shapley Q-Learning

4.1 Definition and Formulation

Shapley-Bellman Optimality Equation. Based on the Bellman optimality equation [15] and the following conditions (the interpretability of which are left to Section 4.2):

C.1. Efficiency of MSV (i.e. the result from Proposition 2);

C.2. $Q_i^{\phi^*}(\mathbf{s}, a_i) = w_i(\mathbf{s}, a_i)\, Q^{\pi^*}(\mathbf{s}, \mathbf{a}) - b_i(\mathbf{s})$, where $w_i(\mathbf{s}, a_i) > 0$ and $b_i(\mathbf{s}) \geq 0$ are bounded and $\sum_{i\in\mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}) = 0$,

we derive *Shapley-Bellman optimality equation* (SBOE) for evaluating the optimal MSQ (an equivalent form to optimal MSV) such that

$$\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = \mathbf{w}(\mathbf{s}, \mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \Big[R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}', a_i) \Big] - \mathbf{b}(\mathbf{s}), \tag{7}$$

where $\mathbf{w}(\mathbf{s}, \mathbf{a}) = [w_i(\mathbf{s}, a_i)]^{\top} \in \mathbb{R}_{+}^{|\mathcal{N}|}$; $\mathbf{b}(\mathbf{s}) = [b_i(\mathbf{s})]^{\top} \in \mathbb{R}_{\geq 0}^{|\mathcal{N}|}$; $\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = [Q_i^{\phi^*}(\mathbf{s}, a_i)]^{\top} \in \mathbb{R}_{\geq 0}^{|\mathcal{N}|}$ and $Q_i^{\phi^*}(\mathbf{s}, a_i)$ denotes the optimal MSQ. If Eq.7 holds, the optimal MSQ is achieved. Moreover, it reveals an implication that for any $\mathbf{s} \in \mathcal{S}$ and $a_i^* = \arg\max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i)$, we have a solution $w_i(\mathbf{s}, a_i^*) = 1/|\mathcal{N}|$ (see Appendix E.4.1). Literally, the assigned credits would be equal and each agent would receive $Q^{\pi^*}(\mathbf{s}, \mathbf{a})/|\mathcal{N}|$ if performing the optimal actions. It is apparent that the efficiency still holds under this situation, which can be interpreted as an extremely fair credit assignment such that the credit to each agent should not be discriminated if all of them perform optimally, regardless of their roles. The equal credit assignment was also revealed by Wang et al. [20] recently from another perspective of analysis. Nevertheless, $w_i(\mathbf{s}, a_i)$ for $a_i \neq \arg\max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i)$ needs to be learned.

Shapley-Bellman Operator. To find an optimal solution described by Eq.7, we now propose an operator called *Shapley-Bellman operator* (SBO), i.e., $\Upsilon : \times_{i\in\mathcal{N}} Q_i^{\phi}(\mathbf{s}, a_i) \mapsto \times_{i\in\mathcal{N}} Q_i^{\phi}(\mathbf{s}, a_i)$, which is defined as follows:

$$\Upsilon\left(\times_{i\in\mathcal{N}} Q_i^{\phi}(\mathbf{s}, a_i)\right) = \mathbf{w}(\mathbf{s}, \mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \Big[R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i) \Big] - \mathbf{b}(\mathbf{s}), \tag{8}$$

where $w_i(\mathbf{s}, a_i) = 1/|\mathcal{N}|$ when $a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i)$. We prove that the optimal joint deterministic policy can be achieved by recursively running SBO in Theorem 1.

Theorem 1. *Shapley-Bellman operator is able to converge to the optimal Markov Shapley Q-value and the corresponding optimal joint deterministic policy when* $\max_{\mathbf{s}} \left\{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \right\} < \frac{1}{\gamma}$.

Shapley Q-Learning. For easy implementation, we conduct transformation for the stochastic approximation of SBO and derive *Shapley Q-learning* (SHAQ) whose TD error is shown as follows:

$$\Delta(\mathbf{s}, \mathbf{a}, \mathbf{s}') = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\, Q_i^{\phi}(\mathbf{s}, a_i), \tag{9}$$

where

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i). \end{cases} \tag{10}$$

Actually, the closed-form expression of $\delta_i(\mathbf{s}, a_i)$ is written as $|\mathcal{N}|^{-1} w_i(\mathbf{s}, a_i)^{-1}$. If inserting the condition that $w_i(\mathbf{s}, a_i) = 1/|\mathcal{N}|$ when $a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i)$ as well as defining $\delta_i(\mathbf{s}, a_i)$ as $\alpha_i(\mathbf{s}, a_i)$ when $a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i)$, Eq.10 is obtained. The term $\mathbf{b}(\mathbf{s})$ is cancelled in Eq.9 thanks to the condition such that $\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}) = 0$. Note that the condition to $w_i(\mathbf{s}, a_i)$ in Theorem 1 should hold for the convergence of SHAQ in implementation (see Appendix E.4.4).

4.2 Validity and Interpretability

In this section, we show the validity of SBOE and the interpretability of SHAQ, i.e., providing the reasons why SBOE is valid to be formulated and SHAQ is an interpretable value factorisation method for the global reward game.

Theorem 2. *The optimal Markov Shapley value is a solution in the Markov core under Markov convex game with the grand coalition.*

Remark 2. *For an arbitrary state $\mathbf{s} \in \mathcal{S}$, by C.2 it is not difficult to check that even if an arbitrary agent i is dummy (i.e., $Q_i^{\phi^*}(\mathbf{s}, a_i) = 0$ for some $i \in \mathcal{N}$), $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ and $Q_j^{\phi^*}(\mathbf{s}, a_j), \forall j \neq i$ would not be zero if $b_i(\mathbf{s}) \neq 0$. If the extreme case happens that for an arbitrary state $\mathbf{s} \in \mathcal{S}$ all agents are dummies, since $\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}) = 0$ we are allowed to set $b_i(\mathbf{s}) = 0, \forall i \in \mathcal{N}$ so that $Q^{\pi^*}(\mathbf{s}, \mathbf{a}) = 0$ and efficiency such that $\max_{\mathbf{a}} Q^{\pi^*}(\mathbf{s}, \mathbf{a}) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i)$ is still valid.*

First, we give a proof for showing that the optimal MSV is a solution in Markov core under the grand coalition, as Theorem 2 shows. Since a solution in Markov core implies the optimal global value (see Remark 5 in Appendix D.2.2), we can conclude that *the optimal MSV can lead to the optimal global value* (a.k.a. social welfare), which links Condition C.1 to Markov core. As a result, *solving SBOE is equivalent to solving Markov core under the grand coalition and SHAQ is actually a learning algorithm that reliably converges to Markov core*. As per the definition in Section 2, we can say that SHAQ leads to the result that no agents have incentives to deviate from the grand coalition, which provides an interpretation of value factorisation for global reward game. Condition C.2 is a condition that *maintains the validity of the relationship between the optimal MSQ and the optimal global Q-value even if there exist dummy agents* (see Remark 2), so that the definition of SBOE is valid for MCG and MSQ in almost every case, which preserves the completeness of the theory.

4.3 Implementations

We now describe a practical implementation of SHAQ for Dec-POMDP [21] (i.e. the global reward game but with partial observations). First, the global state is replaced by the history of each agent to guarantee the optimal deterministic joint policy [21]. Accordingly, Markov Shapley Q-value is denoted as $Q_i^{\phi}(\tau_i, a_i)$, wherein τ_i is a history of partial observations of agent i. Since the paradigm of centralised training decentralised execution (CTDE) [22] is applied, the global state (i.e. $\mathbf{s}$) for $\hat{\alpha}_i(\mathbf{s}, a_i)$ can be obtained during training.

Proposition 3. *Suppose any action marginal contribution can be factorised to the form such that $\Phi_i(\mathbf{s}, a_i | \mathcal{C}_i) = \sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}})\, \hat{Q}_i(\mathbf{s}, a_i)$. With the condition such that*

$$\mathbb{E}_{\mathcal{C}_i \sim Pr(\mathcal{C}_i | \mathcal{N} \setminus \{i\})} \left[\sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \right] = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \\ K \in (0, 1) & a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \end{cases}$$

we have

$$\begin{cases} Q_i^{\phi}(\mathbf{s}, a_i) = \hat{Q}_i(\mathbf{s}, a_i) & a_i = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i)\, Q_i^{\phi}(\mathbf{s}, a_i) = \hat{\alpha}_i(\mathbf{s}, a_i)\, \hat{Q}_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \end{cases} \tag{11}$$

where $\hat{\alpha}_i(\mathbf{s}, a_i) = \mathbb{E}_{\mathcal{C}_i \sim Pr(\mathcal{C}_i | \mathcal{N} \setminus \{i\})} \left[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i}) \right]$ and $\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i}) := \alpha_i(\mathbf{s}, a_i)\, \sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}})$.

Compatible with the decentralised execution, we use only one parametric function $\hat{Q}_i(\tau_i, a_i)$ to directly approximate $Q_i^{\phi}(\tau_i, a_i)$. By inserting Eq.11 into Eq.9, $\delta_i(\mathbf{s}, a_i)$ is transformed into the form as follows:

$$\hat{\delta}_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \\ \hat{\alpha}_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \end{cases} \tag{12}$$

where $\hat{\alpha}_i(\mathbf{s}, a_i) = \mathbb{E}_{\mathcal{C}_i \sim Pr(\mathcal{C}_i|\mathcal{N}\setminus\{i\})}\left[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})\right]$. To solve partial observability, $\hat{Q}_i(\tau_i, a_i)$ is empirically represented as recurrent neural network (RNN) with GRUs [23]. $\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})$ is directly approximated by a parametric function $F_{\mathbf{s}} + 1$ and thus $\hat{\alpha}_i(\mathbf{s}, a_i)$ can be expressed as follows:

$$\hat{\alpha}_i(\mathbf{s}, a_i) = \frac{1}{M}\sum_{k=1}^{M} F_{\mathbf{s}}\left(\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}),\ \hat{Q}_i(\tau_i, a_i)\right) + 1, \tag{13}$$

where $\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}) = \frac{1}{|\mathcal{C}_i^k|}\sum_{j\in\mathcal{C}_i^k}\hat{Q}_j(\tau_j, a_j)$ and $\mathcal{C}_i^k$ is sampled M times from $Pr(\mathcal{C}_i|\mathcal{N}\setminus\{i\})$ (i.e., implemented as Remark 1 suggests) to approximate $\mathbb{E}_{\mathcal{C}_i\sim Pr(\mathcal{C}_i|\mathcal{N}\setminus\{i\})}[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})]$ using Monte Carlo approximation; and $F_{\mathbf{s}}$ is a monotonic function, followed by an absolute activation function, whose weights are generated from hyper-networks w.r.t. the global state. We show that Eq.13 satisfies the condition to $w_i(\mathbf{s}, a_i)$ in Theorem 1 (see Appendix E.6.1), so it is a reliable implementation.

By using the framework of fitted Q-learning [24] to solve large number of states (i.e., could be usually infinite) and plugging in the above designed modules, the practical least-square-error loss function derived from Eq.9 is therefore stated as follows:

$$\min_{\theta,\lambda}\mathbb{E}_{\mathbf{s},\tau,\mathbf{a},R,\tau'}\left[\left(R + \gamma\sum_{i\in\mathcal{N}}\max_{a_i}\hat{Q}_i(\tau_i', a_i; \theta^-) - \sum_{i\in\mathcal{N}}\hat{\delta}_i(\mathbf{s}, a_i; \lambda)\,\hat{Q}_i(\tau_i, a_i; \theta)\right)^2\right], \tag{14}$$

where all agents share the parameters of $\hat{Q}_i(\mathbf{s}, a_i; \theta)$ and $\hat{\alpha}_i(\mathbf{s}, a_i; \lambda)$ respectively; and $\hat{Q}_i(\mathbf{s}', a_i; \theta^-)$ works as the target where θ^- is periodically updated. The general training procedure follows the paradigm of DQN [25], with a replay buffer to store the online collection of agents' episodes. To depict an overview of the algorithm, the pseudo code is shown in Appendix A.

5 Related Work

Value Factorisation in MARL. To deal with the instability during training in global reward game by independent learners [26], the centralised training and decentralised execution (CTDE) [22] was proposed and it became a general paradigm for MARL. Based on CTDE, MADDPG [27] learns a global Q-value that can be regarded as assigning the same credits to all agents during training [13], which may cause the unfair credit assignment [28]. To avoid this problem, VDN [8] was proposed to learn the factorised Q-value, assuming that any global Q-value equals to the sum of decentralised Q-values. Nevertheless, this factorisation may limit the representation of the global Q-value. To mitigate this issue, QMIX [9] and QTRAN [10] were proposed to represent the global Q-value with a richer class w.r.t. decentralised Q-values, based on the assumption (called Individual-Global-Max) of convergence to the optimal joint deterministic policy. Markov Shapley value proposed in this paper belongs to the family of value factorisation, based on the game-theoretical framework called MCG that enjoys the interpretability. From the conventional cooperative games (e.g., network flow game [29], induced subgraph game [30] that can be used for modelling social networks, and facility location game [31]), it is insightful that the coalition introduced in this paper exists. In many scenarios, however, the information of coalition might be unknown. Therefore, the latent coalition is assumed, and we only need to concentrate on the observable information, e.g., the global reward.

Relationship to VDN. By setting $\delta_i(\mathbf{s}, a_i) = 1$ for all state-action pairs, SHAQ degrades to VDN [8]. Although VDN tried to tackle the problem of dummy agents, Sunehag et al. [8] did not give a theoretical guarantee on identifying it. The Markov Shapley value theory proposed in this paper well addresses this issue from both theoretical and empirical aspects. These aspects show that VDN is a subclass of SHAQ. The theoretical framework proposed in this paper answers to why VDN works well in most scenarios but performs poorly in some scenarios (i.e., $\delta_i(\mathbf{s}, a_i) = 1$ in Eq.9 was incorrectly defined over the suboptimal actions).

Relationship to COMA. Compared with COMA [7], each agent i's credit assignment $\bar{Q}_i(\mathbf{s}, a_i)$ is mathematically expressed as follows:

$$\bar{Q}_i(\mathbf{s}, a_i) = \bar{Q}^{\pi}(\mathbf{s}, \mathbf{a}) - \bar{Q}^{\pi_{-i}}(\mathbf{s}, \mathbf{a}_{-i}),$$
$$\bar{Q}^{\pi_{-i}}(\mathbf{s}, \mathbf{a}_{-i}) = \sum_{a_i}\pi_i(a_i|\mathbf{s})\bar{Q}^{\pi}\left(\mathbf{s}, (\mathbf{a}_{-i}, a_i)\right),$$

where subscript $-i$ indicates the agents excluding i. $\bar{Q}_i(\mathbf{s}, a_i)$ can be seen as the action marginal contribution between the grand coalition Q-value and the coalition Q-value excluding the agent i,

under *some permutation to form the grand coalition* wherein agent i is located at the *last position*. The efficiency is obviously violated (i.e., the sum of optimal action marginal contributions defined here is unlikely to be equal to the optimal grand coalition Q-value). In contrast to COMA, SHAQ considers all permutations to form the grand coalition to preserve the efficiency.

Relationship to Independent Learning. Independent learning (e.g. IQL [26]) can be also seen as a special credit assignment, however, the credit assigned to each agent is still with no intuitive interpretation. Mathematically, suppose that $\bar{Q}_i(\mathbf{s}, a_i)$ is the independent Q-value of agent i, we can rewrite it in the form consisting of action marginal contributions such that

$$\bar{Q}_i(\mathbf{s}, a_i) = \mathbb{E}_{\mathcal{C}_i \sim Pr(\mathcal{C}_i | \mathcal{N} \setminus \{i\})} \left[\bar{\Phi}_i(\mathbf{s}, a_i | \mathcal{C}_i) \right].$$

It is intuitive to see that the independent Q-value is a direct approximation of MSQ, ignoring coalition formation, while SHAQ considers coalition formation in approximation. This gives an explanation for why independent learning works well in some cooperative tasks [32]. Nevertheless, it encounters the same issue as in COMA, the loss of properties led by the coalition formation.

Relationship to SQDDPG. We now discuss the relationship between SQDDPG [13] and SHAQ. In terms of algorithms, SQDDPG belongs to policy gradient methods (i.e. an approximation of policy iteration) while SHAQ belongs to value based methods (i.e. an approximation of value iteration). Since policy iteration (with one-step policy evaluation) is equivalent to value iteration [33] (at least under a finite state space and a finite action space), the theory behind SHAQ directly *fills the gap in SQDDPG on theoretical guarantees of convergence to optimal joint policy*. Specifically, the learning procedure of SQDDPG iteratively performs the following two stages:

$$\textbf{Stage 1:} \quad \min_{\theta} \mathbb{E}_{\mathbf{s},\mathbf{a},R,\mathbf{s}'} \left[\left(R + \gamma \sum_{i \in \mathcal{N}} \hat{Q}_i^{\phi}(\mathbf{s}', a_i'; \theta^-) - \sum_{i \in \mathcal{N}} \hat{Q}_i^{\phi}(\mathbf{s}, a_i; \theta) \right)^2 \right].$$

$$\textbf{Stage 2:} \quad \pi_i(\mathbf{s}) \in \arg\max_{a_i} \hat{Q}_i^{\phi}(\mathbf{s}, a_i; \theta).$$

It can be observed that both SQDDPG and SHAQ ideally converge to the same optimal MSQs w.r.t. the optimal actions such that

$$\mathbb{E}_{\mathbf{s},\mathbf{s}'} \left[\left(\max_{\mathbf{a}} R(\mathbf{s}, \mathbf{a}) + \gamma \sum_{i \in \mathcal{N}} \max_{a_i'} \hat{Q}_i^{\phi^*}(\mathbf{s}', a_i') - \sum_{i \in \mathcal{N}} \max_{a_i} \hat{Q}_i^{\phi^*}(\mathbf{s}, a_i) \right)^2 \right] = 0.$$

However, about suboptimal actions, *SQDDPG does not provide any theoretical guarantee*, whereas SHAQ does with specific implementations as shown in Eq.13 to match the theoretical results shown in this paper. Note that this is critical to reliable interpretations of the optimal MSQ w.r.t. suboptimal actions (e.g., for detecting adversarial attacks on controllers if deployed in industry [34]).

6 Experiments

In this section, we show the experimental results of SHAQ on Predator-Prey [17] and various tasks in StarCraft Multi-Agent Challenge (SMAC) [2]. The baselines that we select for comparison are COMA [7], VDN [8], QMIX [9], MASAC [36], QTRAN [10], QPLEX [37] and W-QMIX (including CW-QMIX and OW-QMIX) [35]. The implementation details of our algorithm are shown in Appendix B.1, whereas the implementation of baselines are from [35] [3]. We also compare SHAQ with SQDDPG [13] [4], which is shown in Appendix C.3. For all experiments, we use the ϵ-greedy exploration strategy, where ϵ is annealed from 1 to 0.05. The annealing time steps vary among different experiments. For Predator-Prey, we apply 1 million time steps for annealing, following the setup from [37]. For the easy and hard maps in SMAC, we apply 50k time steps for annealing, the same as that in [18]; while for the super-hard maps in SMAC, we apply 1 million time steps for annealing to obtain more explorations so that more state-action pairs can be visited. About the replay buffer size, we set as 5000 for all algorithms that is the same as that in [35]. To fairly evaluate all algorithms, we run each experiment with 5 random seeds. All graphs showing experimental results are plotted with the

[2] The version that we use in this paper is SC2.4.6.2.69232 rather than the newer SC2.4.10. As reported from [35], the performance is not comparable across versions.

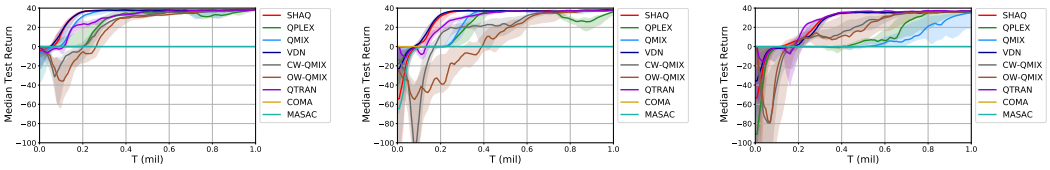
[3] The source code of baseline implementation is from `https://github.com/oxwhirl/wqmix`.

[4] The code of SQDDPG is implemented based on `https://github.com/hsvgbkhgbv/SQDDPG`.

median and 25%-75% quartile shading. About the interpretability of algorithms, we evaluate the algorithms with both both ϵ-greedy policy (i.e., $\epsilon = 0.8$) for obtaining mixed optimal and suboptimal actions and greedy policy for obtaining pure optimal actions. The ablation study of SHAQ is shown in Appendix C.4.

6.1 Predator-Prey

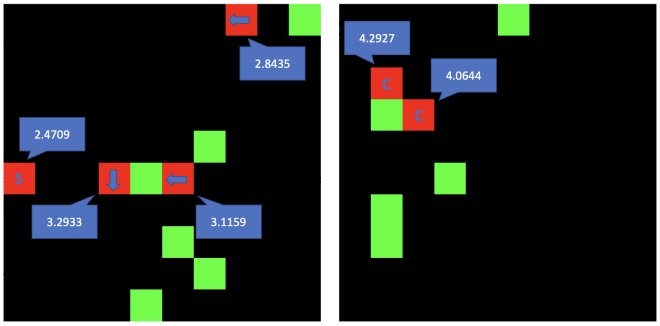
We firstly run the experiments on a partially-observable task called Predator-Prey [17], wherein 8 predators that are feasible to be controlled aim to capture 8 preys with random policies in a 10x10 grid world. Each agent's observation is a 5x5 sub-grid centering around it. If a prey is captured by coordination of 2 agents, predators will be rewarded by 10. On the other hand, each unsuccessful attempt by only 1 agent will be punished by a negative reward p. In this experiment, we study the behaviors of each algorithm under different values of p (that describes different levels of coordination). As [35] reported, only QTRAN and W-QMIX can solve this task, while [37] found that the failure was primarily due to the lack of explorations. As a result, we apply the identical epsilon annealing schedule (i.e. 1 million time steps) adopted in [37].



(a) p=-0.5. (b) p=-1. (c) p=-2.

Figure 1: Median test return for Predator-Prey with different values of p.

Performance Analysis. As Figure 1 shows, SHAQ can always solve the tasks with different values of p. With the epsilon annealing strategy from [37], W-QMIX does not perform as well as reported in [35]. The reason could be its poor robustness to the increased explorations [35] for this environment (see the evidential experimental results in Appendix C.6). The good performance of VDN validates our analysis in Section 5, whereas the performance of QTRAN is surprisingly almost invariant to the value of p. The performances of QPLEX and QMIX become obviously worse when p=-2. The failure of MASAC and COMA could be due to that relative overgeneralisation[5] prevents policy gradient methods from better coordination [39].



(a) SHAQ: ϵ-greedy. (b) SHAQ: greedy.

Figure 2: Visualisation of the evaluation for SHAQ on Predator-Prey: each red square is a controllable agent, whereas each green square indicates a prey. Each agent's factorised Q-value is reported in the bubble in blue and the symbols within the squares indicate the action of each agent (i.e., arrows imply the movement direction, "S" implies staying and "C" implies capturing a prey that is valid only when the agent is around a prey).

Interpretability of SHAQ. To verify that SHAQ possesses the interpretability, we show its credit assignment on Predator-Prey. As we see from Figure 2b, all agents are around and capture a prey, so

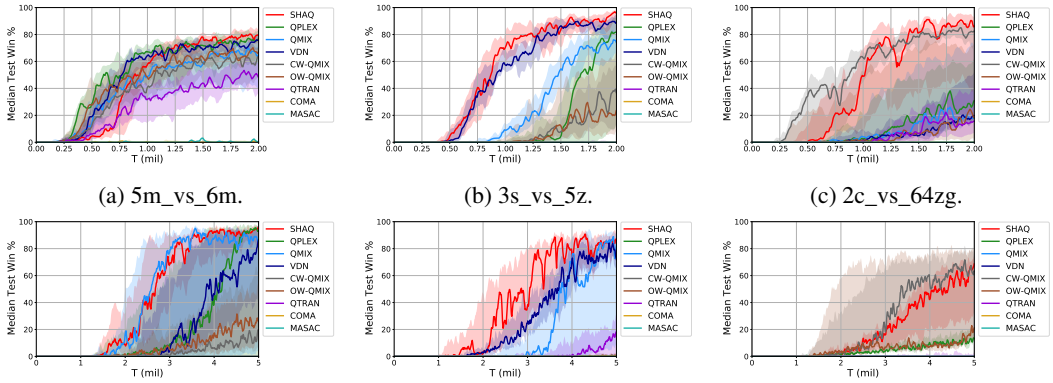
[5]Relative overgeneralisation is a common game theoretic pathology that the suboptimal actions are preferred when matched with arbitrary actions from the collaborating agents [38].

both of them perform the optimal actions and deserve almost the equal optimal credit assignment as 4.2927 and 4.0644, which verifies our theoretical claim. From Figure 2a, it can be seen that two agents are far away from preys, so they receive low credits as 2.4709 and 2.8435. On the other hand, the other two agents are around a prey, but they do not perform the optimal action "capture", so they receive less credits than the two agents in Figure 2b. Nevertheless, they are around a prey, so they perform better than those agents that are far away from preys and receive comparatively greater credits as 3.2933 and 3.1159. The coherent credit assignments in both Figure 2a and 2b implies that the assigned credits reflect agents' contributions (verifying (iii) in Proposition 2) , i.e., each agent receives the credit that is consistent with its decision.

6.2 StarCraft Multi-Agent Challenge

We next evaluate SHAQ on the more challenging SMAC tasks, the environmental settings of which are the same as that in [18]. To broadly compare the performance of SHAQ with baselines, we select 4 easy maps: 8m, 3s5z, 1c3s5z and 10m_vs_11m; 3 hard maps: 5m_vs_6m, 3s_vs_5z and 2c_vs_64zg; and 4 super-hard maps: 3s5z_vs_3s6z, Corridor, MMM2 and 6h_vs_8z. All training is through online data collection. Due to the limited space, we only show partial results in the main part of paper and leave the rest in Appendix C.1.

Performance Analysis. It shows in Figure 3 that SHAQ outperforms all baselines on all maps, except for 6h_vs_8z. On 6h_vs_8z, SHAQ can beat all baselines except for CW-QMIX. VDN performs well on 4 maps but bad on the other 2 maps, which still verifies our analysis in Section 5. QMIX and QPLEX perform well on the most of maps, except for 3s_vs_5z, 2c_vs_64zg and 6h_vs_8z. As for COMA, MADDPG and MASAC, their poor performances could be due to the weak adaptability to challenging tasks. Although QTRAN can theoretically represent the complete class of the global Q-value [10], its complicated learning paradigm could impede the convergence to the value function for challenging tasks and therefore result in the poor performance. Although W-QMIX performs well on some maps, owing to lacking a law on hyperparameter tuning [35] it is difficult to be adapted for all scenarios (see Appendix C.2).



(a) 5m_vs_6m. (b) 3s_vs_5z. (c) 2c_vs_64zg.

(d) 3s5z_vs_3s6z. (e) Corridor. (f) 6h_vs_8z.

Figure 3: Median test win % for hard (a-c), and super-hard (d-f) maps of SMAC.

Interpretability of SHAQ. To further show the interpretability of SHAQ, we also conduct a test on 3m (i.e. a simple task in SMAC). As seen from Figure 4a, Agent 3 faces the direction opposite to enemies, meanwhile, the enemies are out of its attacking range. It can be understood as that Agent 3 does not contribute to the team and thus it is almost a dummy agent. Its MSQ is 0.84 (around 0) that correctly catch the manner of a dummy agent (verifying (i) in Proposition 2). In contrast, Agent 1 and Agent 2 are attacking enemies, while Agent 1 suffers from more attacks (with lower health) than Agent 2. As a result, Agent 1 contributes more than Agent 2 and therefore its MSQ is greater, which implies that the credits reflect agents' contributions (verifying (iii) in Proposition 2). On the other hand, we can see from Figure 4e that with the optimal policies all agents receive almost identical MSQs (verifying the theoretical results in Section 4.1). The above results well verify the theoretical analysis that we deliver before.

(a) SHAQ: ϵ-greedy. (b) VDN: ϵ-greedy. (c) QMIX: ϵ-greedy. (d) QPLEX: ϵ-greedy.

(e) SHAQ: greedy. (f) VDN: greedy. (g) QMIX: greedy. (h) QPLEX: greedy.

Figure 4: Visualisation of the test for SHAQ and baselines on 3m in SMAC: each colored circle is the centered attacking range of a controllable agent (in red), and each agent's factorised Q-value is reported on the right. We mark the direction that each agent face by an arrow for clearness.

To justify that the MSQs learned by SHAQ are non-trivial, we also show the results of VDN, QMIX and QPLEX. It is surprising that the Q-values of these baselines are also almost identical among agents for the optimal actions (however, the property disappears in more complicated scenarios as shown in Appendix C.5 while the property of SHAQ is still valid). Since VDN is a subclass of SHAQ and possesses the same form of loss function for optimal actions, it is reasonable that it obtains the similar results to SHAQ. As for the suboptimal actions, VDN does not possess an explicit interpretation as SHAQ due to the incorrect definition of $\delta_i(\mathbf{s}, a_i) = 1$ over suboptimal actions (verifying the statement in Section 5). The values of QMIX and QPLEX are difficult to be explained.

7 Conclusion

Summary. This paper generalises Shapley value to Markov convex game, called Markov Shapley value. Markov Shapley value inherits a number of properties: (i) identifiability of dummy agents; (ii) efficiency; (iii) reflecting the contribution and (iv) symmetry. Based on Property (ii), we derive Shapley-Bellman optimality equation, Shapley-Bellman operator and SHAQ. We prove that solving Shapley-Bellman optimality equation is equivalent to solving the Markov core (i.e., no agent has incentives to deviate from the grand coalition). Markov convex game with the grand coalition is equivalent to global reward game [13], wherein Markov Shapley value plays the role of value factorisation. Since SHAQ is a stochastic approximation of Shapley-Bellman operator that is proved to solve Shapley-Bellman optimality equation, global reward game with value factorisation becomes valid standing by the cooperative game theoretical framework (i.e. solving Markov core). Property (i) and (iii) in Proposition 2 are demonstrated in the experiments showing the interpretability of SHAQ.

Limitation and Future Work. The value of Markov convex game is not limited to solving problems with the grand coalition, though in this paper we design SHAQ that only focuses on the scenario with the grand coalition. By removing the condition of supermodularity (see Eq.1), this framework can be used to study more general coalition games where different coalitions of agents as units may compete/cooperate with each other. Since the grand coalition and Markov Shapley value is not a solution in Markov core yet, the learning process becomes more complicated to converge to Markov core. A possible research direction in future is to investigate dynamically forming the coalition structure and conducting credit assignments simultaneously.

Acknowledgements

This work is sponsored by the Engineering and Physical Sciences Research Council of UK (EPSRC) under awards EP/S000909/1. Tae-Kyun Kim is partly sponsored by KAIA grant (22CTAP-C163793-02, MOLIT), NST grant (CRC 21011, MSIT), KOCCA grant (R2022020028, MCST) and the Samsung Display corporation. Yuan Zhang is sponsored by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 953348 (ELO-X).

References

[1] Tamás Keviczky, Francesco Borrelli, Kingsley Fregene, Datta Godbole, and Gary J Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on control systems technology*, 16(1):19–33, 2007.

[2] Arne Schuldt. Multiagent coordination enabling autonomous logistics. *KI-Künstliche Intelligenz*, 26(1):91–94, 2012.

[3] Jianhong Wang, Wangkun Xu, Yunjie Gu, Wenbin Song, and Tim Green. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Sainbayar Sukhbaatar, arthur szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems 29*, pages 2244–2252. Curran Associates, Inc., 2016.

[5] Shayegan Omidshafiei, Dong-Ki Kim, Miao Liu, Gerald Tesauro, Matthew Riemer, Christopher Amato, Murray Campbell, and Jonathan P How. Learning to teach in cooperative multiagent reinforcement learning. *arXiv preprint arXiv:1805.07830*, 2018.

[6] Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019.

[7] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[8] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

[9] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4292–4301. PMLR, 2018.

[10] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Hostallero, and Yung Yi. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5887–5896. PMLR, 2019.

[11] Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. Learning implicit credit assignment for multi-agent actor-critic. *arXiv preprint arXiv:2007.02529*, 2020.

[12] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

[13] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7285–7292, Apr 2020.

[14] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5 (6):1–168, 2011.

[15] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952.

[16] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[17] Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *International Conference on Machine Learning*, pages 980–991. PMLR, 2020.

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

[19] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28): 307–317, 1953.

[20] Jianhao Wang, Zhizhou Ren, Beining Han, Jianing Ye, and Chongjie Zhang. Towards understanding linear value decomposition in cooperative multi-agent q-learning. *arXiv preprint arXiv:2006.00587*, 2020.

[21] Frans A Oliehoek. Decentralized pomdps. In *Reinforcement Learning*, pages 471–503. Springer, 2012.

[22] Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[23] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[24] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[26] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998(746-752):2, 1998.

[27] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390, 2017.

[28] David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. In *Modeling complexity in economic and social systems*, pages 355–369. World Scientific, 2002.

[29] Ehud Kalai and Eitan Zemel. Generalized network problems yielding totally balanced games. *Operations Research*, 30(5):998–1008, 1982.

[30] Xiaotie Deng and Christos H Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of operations research*, 19(2):257–266, 1994.

[31] Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999.

[32] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[33] Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[34] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic control*, 59(6): 1454–1467, 2014.

[35] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[36] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 2961–2970. PMLR, 2019.

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

[38] Ermo Wei and Sean Luke. Lenient learning in independent-learner stochastic cooperative games. *The Journal of Machine Learning Research*, 17(1):2914–2955, 2016.

[39] Ermo Wei, Drew Wicke, David Freelan, and Sean Luke. Multiagent soft q-learning. In *2018 AAAI Spring Symposium Series*, 2018.

[40] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[41] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[42] Richard H. Byrd, Gillian M. Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical Programming*, 134(1):127–155, 2012. doi: 10.1007/s10107-012-0572-5.

[43] Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 2305–2313. Curran Associates, Inc., 2015.

[44] Lloyd S Shapley. Cores of convex games. *International journal of game theory*, 1(1):11–26, 1971.

[45] Harold Garth Dales, H Garth Dales, Pietro Aiena, Jörg Eschmeier, Kjeld Laursen, and George A Willis. *Introduction to Banach algebras, operators, and harmonic analysis*, volume 57. Cambridge University Press, 2003.

[46] Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. math*, 3(1):133–181, 1922.

[47] Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

[48] Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix F.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix E.1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix E.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Algorithm of Shapley Q-learning

In this section, we present the pseudo code of Shapley Q-learning in Algorithm 1. The general paradigm can be divided into such parts: (1) collecting samples through ϵ-greedy strategy and store the collected samples to a replay buffer for training; (2) sampling a batch of episodes of samples from the replay buffer; (3) calculating $\hat{Q}_i(\tau_i^{t+1}, a_i^{t+1}; \theta^-)$, $\hat{\alpha}_i(\mathbf{s}^k, a_i^k; \lambda)$ and $\hat{Q}_i(\tau_i^t, a_i^t; \theta)$; and (4) constructing a loss of Shapley Q-learning and updating parameters to minimise the loss.

Algorithm 1 Shapley Q-learning

1: Initialise a set of agents $\mathcal{N}$ and set $N = |\mathcal{N}|$
2: Initialise $\hat{Q}_i(\tau_i, a_i; \theta)$ with the shared parameters among agents
3: Initialise $\hat{\alpha}_i(\mathbf{s}, a_i; \lambda)$ with the shared parameters among agents
4: Initialise $\hat{Q}_i(\tau_i, a_i; \theta^-)$ by copying $\hat{Q}_i(\tau_i, a_i; \theta)$ with the shared parameters among agents
5: Initialise a replay buffer $\mathcal{B}$
6: **repeat**
7: Initialise a container $\mathcal{E}$ for storing an episode
8: Observe an initial global state $\mathbf{s}^1$ and each agent's partial observation o_i^1 from an environment
9: **for** t=1:T **do**
10: Get $\tau_i^t = (o_i^m)_{m=1:t}$ for each agent
11: For each agent i, select an action

$$a_i^t = \begin{cases} \text{a random action} & \text{with probability } \epsilon \\ \arg\max_{a_i} \hat{Q}_i^*(\tau_i^t, a_i; \theta) & \text{otherwise} \end{cases}$$

12: Execute a_i^t of each agent to get the global reward R^t, $\mathbf{s}^{t+1}$ and each agent's o_i^{t+1}
13: Store $\left\langle \mathbf{s}^t, (o_i^t)_{i=1:N}, (a_i^t)_{i=1:N}, R^t, \mathbf{s}^{t+1}, (o_i^{t+1})_{i=1:N} \right\rangle$ to $\mathcal{E}$
14: **end for**
15: Store $\mathcal{E}$ to $\mathcal{B}$
16: Sample a batch of episodes with batch size B from $\mathcal{B}$
17: **for** each sampled episode **do**
18: **for** k=1:T **do**
19: Get each transition $\left\langle \mathbf{s}^k, (o_i^k)_{i=1:N}, (a_i^k)_{i=1:N}, R^k, \mathbf{s}^{k+1}, (o_i^{k+1})_{i=1:N} \right\rangle$
20: For each agent i, get $\tau_i^k = (o_i^m)_{m=1:k}$
21: For each agent i, calculate $\hat{Q}_i(\tau_i^k, a_i^k; \theta)$
22: For each agent i, calculate $\alpha_i(\mathbf{s}^k, a_i^k; \lambda)$ by Algorithm 2
23: For each agent i, calculate $\delta_i(\mathbf{s}^k, a_i^k; \lambda)$ as follows:

$$\hat{\delta}_i(\mathbf{s}^k, a_i^k; \lambda) = \begin{cases} 1 & a_i^k = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}^k, a_i; \theta) \\ \hat{\alpha}_i(\mathbf{s}^k, a_i^k; \lambda) & a_i^k \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}^k, a_i; \theta) \ \text{(via Algorithm 2)} \end{cases}$$

24: For each agent i, get $\tau_i^{k+1} = (o_i^m)_{m=1:k+1}$
25: For each agent i, get a_i^{k+1} by $\arg\max_{a_i} \hat{Q}_i(\tau_i^{k+1}, a_i; \theta)$
26: For each agent i, calculate $\hat{Q}_i(\tau_i^{k+1}, a_i^{k+1}; \theta^-)$
27: **end for**
28: **end for**
29: Construct a loss as follows:

$$\min_{\theta,\lambda} \frac{1}{B} \sum_{k=1}^{B} \Big[\Big(R^k + \gamma \sum_{i \in \mathcal{N}} \max_{a_i^k} \hat{Q}_i(\tau_i^{k+1}, a_i^{k+1}; \theta^-) - \sum_{i \in \mathcal{N}} \hat{\delta}_i(\mathbf{s}^k, a_i^k; \lambda)\, \hat{Q}_i(\tau_i^k, a_i^k; \theta) \Big)^2 \Big]$$

30: Update θ and λ through the above loss
31: Periodically update θ^- by copying θ
32: **until** $\hat{Q}_i(\tau_i, a_i; \theta)$ converges

Implementation of Sampling from $Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ (Line 4 in Algorithm 2). As introduced in Remark 1, the analytic form of $Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ is $\frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}$ that is actually the occurrence frequency of correlated coalition $\mathcal{C}_i$. Since each coalition is formed by different permutations, it can be instead sampled from permutations directly with uniform distribution where $\frac{1}{|\mathcal{N}|!}$ is as the probability distribution over each permutation. It is not difficult to find that these two sampling strategy induce the same probability distribution for obtaining $\mathcal{C}_i$, so they are equivalent. In practice, we sample multiple permutations (saying M) from the uniform distribution in parallel. From each sampled permutation, we extract the the relevant $\mathcal{C}_i$ for each agent i. Afterwards, to each agent i, M coalitions are obtained to calculate $\hat{\alpha}_i(\mathbf{s}, a_i)$.

Algorithm 2 Calculating $\hat{\alpha}_i(\mathbf{s}, a_i)$

Input: s, $\left(\hat{Q}_i(\tau_i, a_i; \theta)\right)_{i=1:N}$, M
Output: $\left(\hat{\alpha}_i(\mathbf{s}, a_i)\right)_{i=1:N}$
for each agent i **do**
  Sample M preceding coalitions $\mathcal{C}_i^k \sim Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$
  for k=1:M **do**
    Get $\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}) = \frac{1}{|\mathcal{C}_i^k|}\sum_{j\in\mathcal{C}_i^k}\hat{Q}_j(\tau_j, a_j)$
  end for
  Get $\hat{\alpha}_i(\mathbf{s}, a_i) = \frac{1}{M}\sum_{k=1}^{M} F_{\mathbf{s}}\Big(\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}),\ \hat{Q}_i(\tau_i, a_i)\Big) + 1$
end for

B Experimental Setups

B.1 Implementation Details of Shapley Q-learning

We now provide the additional implementation details that are omitted from the main part of paper. First, $F_s(\cdot,\cdot)$ is a 3-layer network (consecutively with two affine transformation and an activation of absolute), where the hidden-layer dimension is 32. The parameters of each affine transformation are generated by hyper-networks [40] with input as the global state, whose details are shown in Table 1. The architecture of each agent's Q-value is a RNN with GRUs cell [23], whose hidden-layer dimension is 64. The input dimension is state dimension and the output dimension is action dimension.

Table 1: Table of specifications for $F_s(\cdot,\cdot)$.

NETWORK	STRUCTURE
1ST WEIGHT MATRIX	[LINEAR(STATE_DIM, 64), RELU, LINEAR(64, 32*2), ABSOLUTE]
1ST BIAS	[LINEAR(STATE_DIM, 64)]
2ND WEIGHT MATRIX	[LINEAR(STATE_DIM, 64), RELU, LINEAR(64, 32), ABSOLUTE]
2ND BIAS	[LINEAR(STATE_DIM, 32), RELU, LINEAR(32, 1)]

Taking the lessons of training two coupling modules from GANs [41], we take separate learning rates for $\hat{\alpha}_i(\mathbf{s}, a_i)$ and $\hat{Q}_i(\mathbf{s}, a_i)$. The learning rate for $\hat{Q}_i(\mathbf{s}, a_i)$ is fixed at 0.0005 for all tasks. Nevertheless, the learning rate for $\hat{\alpha}_i(\mathbf{s}, a_i)$ is dependent on the number of controllable agents. We use RMSProp optimizer for training in all tasks. All models are implemented in PyTorch 1.4.0 and each experiment is run on Nvidia GeForce RTX 2080Ti for 4 to 26 hours with a single process of environment.

B.2 Hyperparameters of Baselines

The hyperparameters of all baselines except for SQDDPG [13] are consistent with Rashid et al. [35] and Wang et al. [37]. The hyperparamers of SQDDPG are shown as follows: (1) The policy network is consistent with the other baselines, while the critic network is with 3 hidden layers and each layer is with 64 neurons. (2) The policy network is updated every 2 time steps, while the critic network is updated each time step. (3) The multiplier of the entropy of policy is 0.005. The rest of settings are identical with other baselines.

B.3 Predator-Prey for Modelling Relative Overgeneralisation

We give the experimental setups of Predator-Prey [17] in Table 2.

B.4 StarCraft Multi-Agent Challenge

The StarCraft Multi-Agent Challenge (SMAC) [18] is a popular testbed for multi-agent reinforcement learning (MARL) algorithms. The main difficulties are (1) challenging dynamics, (2) partial observability and (3) high-dimensional observation space. During training, both the global state of the environment and each agent's local observation are able to be obtained; however, during execution,

Table 2: Table of experimental setups of Predator-Prey.

HYPERPARAMETERS	VALUE	DESCRIPTION
BATCH SIZE	32	THE NUMBER OF EPISODES FOR EACH UPDATE
DISCOUNT FACTOR γ	0.99	THE IMPORTANCE OF FUTURE REWARDS
REPLAY BUFFER SIZE	5,000	THE MAXIMUM NUMBER OF EPISODES TO STORE IN MEMORY
EPISODE LENGTH	200	MAXIMUM TIME STEPS PER EPISODE
TEST EPISODE	16	THE NUMBER OF EPISODES FOR EVALUATING THE PERFORMANCE
TEST INTERVAL	10,000	THE TIME STEP FREQUENCY FOR EVALUATING THE PERFORMANCE
EPSILON START	1.0	THE START EPSILON ϵ VALUE FOR EXPLORATION
EPSILON FINISH	0.05	THE FINAL EPSILON ϵ VALUE FOR EXPLORATION
EXPLORATION STEP	1,000,000	THE NUMBER OF STEPS FOR LINEARLY ANNEALING ϵ
MAX TRAINING STEP	1,000,000	THE NUMBER OF TRAINING STEPS
TARGET UPDATE INTERVAL	200	THE UPDATE FREQUENCY FOR TARGET NETWORK
LEARNING RATE	0.0001	THE LEARNING RATE FOR $\delta_i(\mathbf{s}, a_i)$
α FOR W-QMIX VARIANTS	0.1	THE WEIGHT FOR CW-QMIX AND OW-QMIX
SAMPLE SIZE	10	THE SAMPLE SIZE FOR COALITION SAMPLING

Table 3: Introduction of maps and characters in SMAC.

MAP NAME	ALLY UNITS	ENEMY UNITS	CATEGORIES
3S5Z	3 STALKERS & 5 ZEALOTS	3 STALKERS & 5 ZEALOTS	EASY
1C3S5Z	1 COLOSSI & 3 STALKERS & 5 ZEALOTS	1 COLOSSI & 3 STALKERS & 5 ZEALOTS	EASY
8M	8 MARINES	8 MARINES	EASY
10M_VS_11M	10 MARINES	11 MARINES	EASY
5M_VS_6M	5 MARINES	6 MARINES	HARD
3S_VS_5Z	3 STALKERS	5 ZEALOTS	HARD
2C_VS_64ZG	2 COLOSSI	64 ZERGLINGS	HARD
3S5Z_VS_3S6Z	3 STALKERS & 5 ZEALOTS	3 STALKERS & 6 ZEALOTS	SUPER-HARD
MMM2	1 MEDIVAC, 2 MARAUDERS & 7 MARINES	1 MEDIVAC, 3 MARAUDERS & 8 MARINES	SUPER-HARD
6H_VS_8Z	6 HYDRALISKS	8 ZERGLINGS	SUPER-HARD
CORRIDOR	6 ZEALOTS	24 ZERGLINGS	SUPER-HARD

only each agent's local observation can be observed. For this reason, SMAC fits the centralised training and decentralised execution (CTDE) paradigm. In each micromanagement task, the ally units are controlled by agents and the enemy units are controlled by the built-in game AI. The agents need to learn a strategy to solve some challenging combat scenarios and defeat their opponents with maximum win rate.

In this paper, we evaluate the proposed SHAQ on 11 typical combat scenarios in SMAC that can be classified into three categories: easy (8m, 3s5z, 1c3s5z and 10m_vs_11m), hard (5m_vs_6m, 3s_vs_5z and 2c_vs_64zg), and super-hard (3s5z_vs_3s6z, Corridor, MMM2 and 6h_vs_8z). More details of these tasks are provided in Table 3. The specific experimental setups for SMAC are shown in Table 4 and 5.

Table 4: Table of experimental setups for SMAC.

HYPERPARAMETERS	EASY	HARD	SUPER HARD	DESCRIPTION
BATCH SIZE	32	32	32	THE NUMBER OF EPISODES FOR EACH UPDATE
DISCOUNT FACTOR γ	0.99	0.99	0.99	THE IMPORTANCE OF FUTURE REWARDS
REPLAY BUFFER SIZE	5,000	5,000	5,000	THE MAXIMUM NUMBER OF EPISODES TO STORE IN MEMORY
MAX TRAINING STEP	2,000,000	2,000,000	5,000,000	THE NUMBER OF TRAINING STEPS
TEST EPISODE	32	32	32	THE NUMBER OF EPISODES FOR EVALUATION
TEST INTERVAL	10,000	10,000	10,000	THE TIME STEP FREQUENCY FOR EVALUATING THE PERFORMANCE
EPSILON START	1.0	1.0	1.0	THE START EPSILON ϵ VALUE FOR EXPLORATION
EPSILON FINISH	0.05	0.05	0.05	THE FINAL EPSILON ϵ VALUE FOR EXPLORATION
EXPLORATION STEP	50,000	50,000	1,000,000	THE NUMBER OF STEPS FOR LINEARLY ANNEALING ϵ
TARGET UPDATE INTERVAL	200	200	200	THE UPDATE FREQUENCY FOR TARGET NETWORK
α FOR OW-QMIX	0.5	0.5	0.5	THE WEIGHT FOR OW-QMIX
α FOR CW-QMIX	0.75	0.75	0.75	THE WEIGHT FOR CW-QMIX
SAMPLE SIZE	10	10	10	THE SAMPLE SIZE FOR COALITION SAMPLING

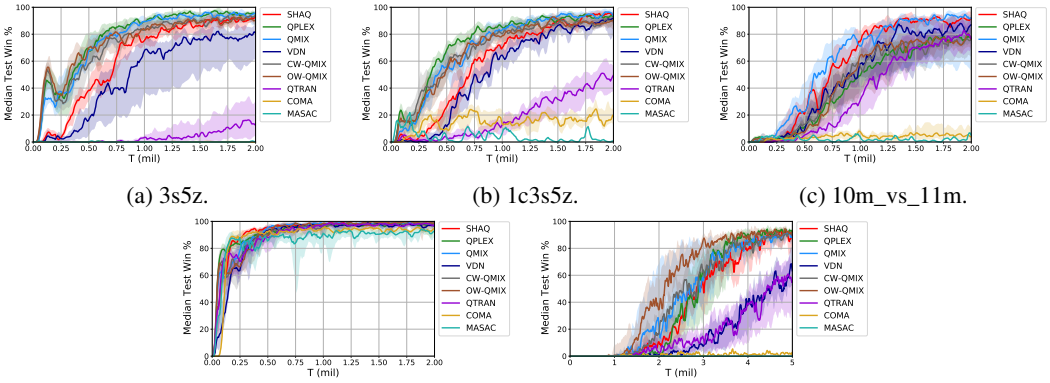
Table 5: The learning rate for training $\hat{\alpha}_i(\mathbf{s}, a_i)$ of SHAQ for various maps in SMAC.

MAP NAME	NUMBER OF AGENTS	LEARNING RATE FOR $\hat{\alpha}_i(\mathbf{s}, a_i)$
2C_VS_64ZG	2	0.002
3S_VS_5Z	3	0.001
5M_VS_6M	5	0.0005
6H_VS_8Z	6	0.0005
CORRIDOR	6	0.0005
8M	8	0.0003
3S5Z	8	0.0003
3S5Z_VS_3S6Z	8	0.0003
1C3S5Z	9	0.0002
10M_VS_11M	10	0.0001
MMM2	10	0.0001

C Extra Experimental Results

C.1 Experimental Results on Extra SMAC Maps

To thoroughly compare the performance of SHAQ with baselines, we also run experiments on 5 extra maps in SMAC as Figure 5 shows. 8m, 3s5z, 1c3s5z and 10m_vs_11m are an easy maps and MMM2 is a super-hard map. The strategy of epsilon annealing is consistent with the previous experiments for SMAC. It is obvious that SHAQ also performs generally well on these 5 maps.



(a) 3s5z. (b) 1c3s5z. (c) 10m_vs_11m.

(d) 8m. (e) MMM2.

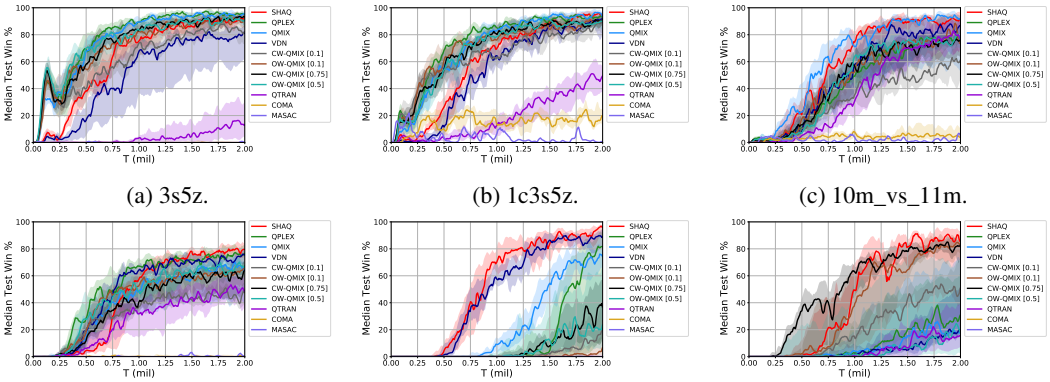
Figure 5: Median test win % for 5 extra maps in SMAC.

C.2 Extra Experimental Results on W-QMIX with $\alpha = 0.1$

To show the significance of tuning α for W-QMIX, we also run W-QMIX with $\alpha = 0.1$ in addition to the best α reported in [35]. We can observe from Figure 6 that the performances of W-QMIX are not comparatively identical for each choice of α. As a result, W-QMIX suffers from the separate tuning of α for each scenario. Unfortunately, Rashid et al. [35] did not provide an empirical law for selecting α, while SHAQ enjoys an empirical law to select $\hat{\alpha}_i(\mathbf{s}, a_i)$ as Figure 8b shows.

C.3 Comparison with SQDDPG

To emphasize the improvement of SHAQ from SQDDPG [13], we exclusively compare these two algorithms on 3 maps in SMAC. As Figure 7 shows, the performance of SHAQ surpasses that of SQDDPG on all 3 maps, while SQDDPG can only learn on the simplest map 3m. The most possible reason for the failure of SQDDPG to complicated tasks is its sample complexity inefficiency for permutations of agents as discussed in Section 5 that leads to the difficulty in learning. Apparently, the implementation of coalition invariance of SHAQ mitigates this weakness so that it is able to solve

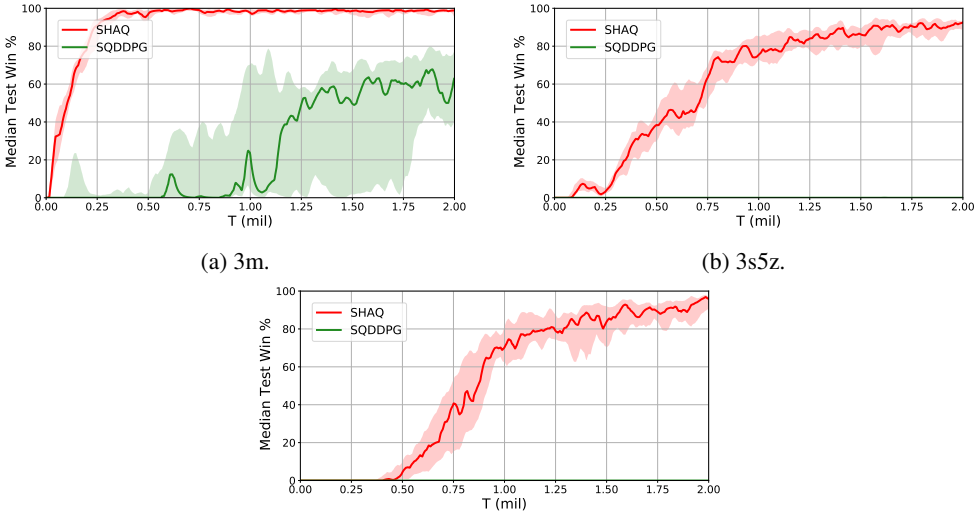


(a) 3s5z. (b) 1c3s5z. (c) 10m_vs_11m.

(d) 5m_vs_6m. (e) 3s_vs_5z. (f) 2c_vs_64zg.

Figure 6: Median test win % for easy (1st row) and hard (2nd row) maps of SMAC for W-QMIX with different α.

more challenging tasks. We also show the results for SQDDPG on Predator-Prey with the same setups (i.e., the epsilon annealing steps are 1 mil), as Figure 10a shows. It is apparent that SHAQ can still outperform SQDDPG.



(a) 3m. (b) 3s5z.

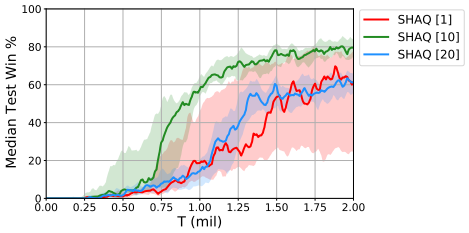
(c) 3s_vs_5z.

Figure 7: Median test win % for 3 maps of SMAC to compare SHAQ with SQDDPG.

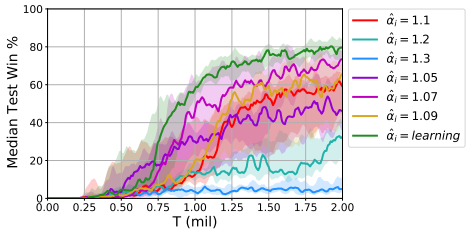
C.4 Ablation Study

We also conduct ablation study of SHAQ, such as the sample size M for approximating $\hat{\alpha}_i(\mathbf{s}, a_i)$, the empirical selection law on the learning rate of $\hat{\alpha}_i(\mathbf{s}, a_i)$, and the demonstration of the necessity of learning $\hat{\alpha}_i(\mathbf{s}, a_i)$ rather than manual setting. These results show that SHAQ is an easy-to-use algorithm that is potential to be applied to other scenarios with less efforts on tuning hyperparameters.

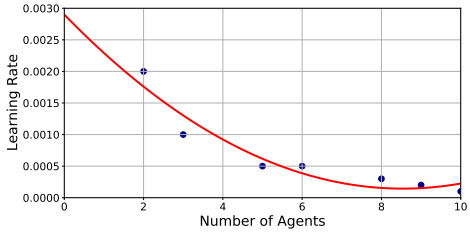
Sample Size M for Approximating $\hat{\alpha}(\mathbf{s}, a_i)$. To study the impact of sample size M on the performance of SHAQ, we conduct an ablation study as Figure 8a shows. We observe that the small M is able to achieve fast convergence rate but with high variance, while the large M is with low variance but comparatively slow convergence rate. The observations are consistent with the conclusions from stochastic optimisation [42, 43]. As a result, we select M = 10 in practice, to trade off between convergence rate and variance.



(a) Comparison among different values of M on 5m_vs_6m. The $[\cdot]$ indicates the value of M.



(b) Comparison between the manually preset and the learned $\hat{\alpha}_i(s, a_i)$ on 5m_vs_6m.



(c) Relationship between learning rate of $\hat{\alpha}_i(s, a_i)$ and the number of agents (in red curve).

Figure 8: The figures of 3 ablation studies of SHAQ on SMAC.

An Empirical Law on Selecting the Learning Rate of $\hat{\alpha}_i(s, a_i)$. To provide an empirical law on selecting the learning rate of $\hat{\alpha}_i(s, a_i)$, we statistically fit a curve of the learning rate w.r.t. the number of controllable agents by the experimental results on SMAC that is shown in Figure 8c. It is seen that the learning rate of $\hat{\alpha}_i(s, a_i)$ is generally negatively related to the number of agents. In other words, as the number of agents grows the learning rate of $\hat{\alpha}_i(s, a_i)$ is recommended to be smaller. For example, if the number of agents is more than 10, the learning rate of $\hat{\alpha}_i(s, a_i)$ is recommended to be 0.0001 as the guidance from Figure 8c.

The Necessity of Learning $\hat{\alpha}_i(\mathbf{s}, a_i)$. Some readers may be concerned about the necessity of learning $\hat{\alpha}_i(\mathbf{s}, a_i)$. To answer this question, we study the necessity of learning $\hat{\alpha}_i(\mathbf{s}, a_i)$ on 5m_vs_6m. Since the learned $\hat{\alpha}_i(\mathbf{s}, a_i)$ finally converges to 1.1029, we grid search the fixed values of $\hat{\alpha}_i(\mathbf{s}, a_i)$ around this number. As Figure 8b shows, $\hat{\alpha}_i(\mathbf{s}, a_i)$ with manually preset fixed value cannot work as well as the learned $\hat{\alpha}_i(\mathbf{s}, a_i)$. Therefore, we demonstrate the necessity of learning $\hat{\alpha}_i(\mathbf{s}, a_i)$ here.

C.5 More Visualisation for Interpretability of SHAQ

(a) SHAQ: ϵ-greedy. (b) VDN: ϵ-greedy. (c) QMIX: ϵ-greedy. (d) QPLEX: ϵ-greedy.

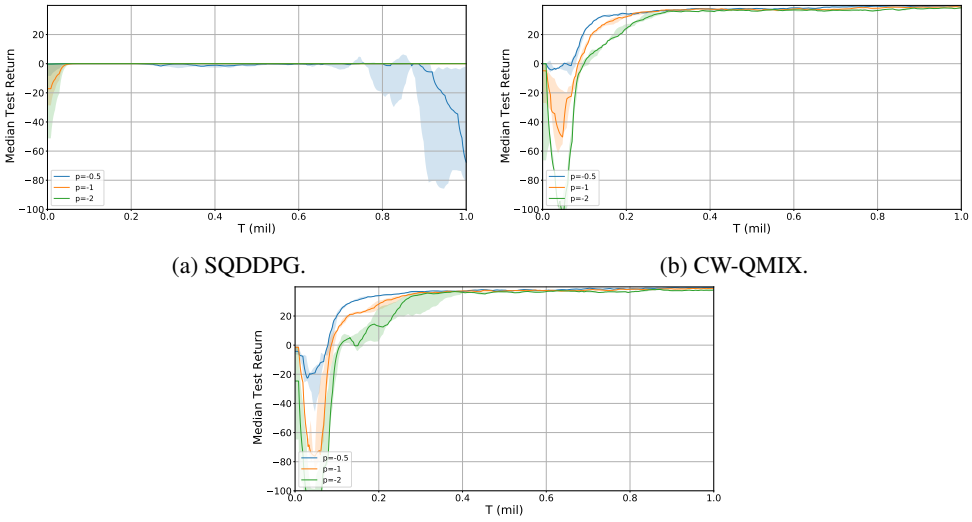
(e) SHAQ: greedy. (f) VDN: greedy. (g) QMIX: greedy. (h) QPLEX: greedy.

Figure 9: Visualisation of the evaluation for SHAQ and baselines on 3s5z_vs_3s6z in SMAC: each colored circle is the centered attacking range of a controllable agent (in red), and each agent's factorised Q-value is reported on the right. We mark the direction that each moving agent face by an arrow.

To verify our theoretical results more firmly, we show the Q-values on a more complicated scenario in SMAC, i.e. 3s5z_vs_3s6z during test in Figure 9. First, we take a look into the optimal actions. SHAQ can still demonstrate the equal credit assignment as we claimed before. Unfortunately, VDN does not explicitly show equal credit assignment. The possible reason is that part of parameters of Q-value are shared between optimal actions and suboptimal actions. Therefore, the parametric effects of the mistakes conducted on suboptimal actions to the optimal actions by VDN during learning may be exaggerated when the number of agents increases. About QMIX and QPLEX, the Q-values of optimal actions are difficult to be interpreted in this complicated scenario. For both algorithms, the agent who is responsible for kiting [6] (i.e. Agent 3 for QMIX and Agent 2 for QPLEX) receives the lowest credit, however, it is an important role to the team in a combat tactic. Next, we focus on the demonstration of the suboptimal actions. As for SHAQ, Agent 1 and Agent 3 are participating into the battle, so deserving almost the equal credit assignment. However, Agent 2 drops teammates and escapes from the center of battle, so it contributes almost nothing to the team. As a result, it can be seen as a dummy agent and thus obtains the credit near 0. This again agrees with our theoretical analysis. About VDN, it coincidentally receives near 0 for the dummy agent (i.e. Agent 3) in this scenario. Nevertheless, the low credit assignments to the other 2 agents who participate in the battle are difficult to be interpreted. About QMIX, the agents who participate in the battle (i.e. Agent 2 and Agent 3) receive the lowest credits, while the agent (i.e. Agent 1) who escapes from the battle receives the highest credit. For QPLEX, the agents' behaviours are difficult to be interpreted.

C.6 Extra Experimental Results of Predator-Prey

In Figure 10b and Figure 10c, we show the results of W-QMIX with the annealing steps as 50k to support that the poor performance of W-QMIX on Predator-Prey is due to its poor robustness to the increased explorations.



(a) SQDDPG.

(b) CW-QMIX.

(c) OW-QMIX.

Figure 10: Median test return for SQDDPG and W-QMIX (including OW-QMIX and CW-QMIX) on Predator-Prey.

D Additional Background

D.1 Value Factorisation in MARL

Although there are lots of works on value factorisation in MARL, most of them are based on an assumption called Individual-Global-Max (IGM) [10] that is defined in Definition 3.

[6] https://en.wikipedia.org/wiki/Glossary_of_video_game_terms.

Definition 3. *For a joint Q-value $Q^{\pi}(\mathbf{s}, \mathbf{a})$ with a deterministic policy, if the following equation is assumed to hold such that*

$$\arg\max_{\mathbf{a}} Q^{\pi}(\mathbf{s}, \mathbf{a}) = \left(\arg\max_{a_i} Q_i(\mathbf{s}, a_i) \right)_{i=1,2,...,|\mathcal{N}|}, \tag{15}$$

then we say that $\left(Q_i(\mathbf{s}, a_i) \right)_{i=1,2,...,|\mathcal{N}|}$ satisfies Individual-Global-Max (IGM) and $Q^{\pi}(\mathbf{s}, \mathbf{a})$ can be factorised by $\left(Q_i(\mathbf{s}, a_i) \right)_{i=1,2,...,|\mathcal{N}|}$.

There are 3 popular frameworks that are followed by most of works implementing the IGM, called VDN [8], QMIX [9] and QTRAN [10].

VDN. VDN linearly factorises a global value function such that

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = \sum_{i \in \mathcal{N}} Q_i(\mathbf{s}, a_i), \tag{16}$$

so that Eq.15 holds.

QMIX. QMIX learns a monotonic mixing function $f_{\mathbf{s}} : \times_{i \in \mathcal{N}} Q_i(\mathbf{s}, a_i) \times \mathbf{s} \mapsto \mathbb{R}$ to implement the factorisation such that

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = f_{\mathbf{s}} \left(Q_1(\mathbf{s}, a_1), ..., Q_{|\mathcal{N}|}(\mathbf{s}, a_{|\mathcal{N}|}) \right), \tag{17}$$

so that Eq.15 holds. Although QMIX has a richer functional class of factorisation than that of VDN, it meets a problem that $\max_{\mathbf{a}} Q^{\pi}(\mathbf{s}, \mathbf{a}) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i(\mathbf{s}, a_i)$ does not necessarily hold, which may lead to the bias on Q-value estimation [10] and affect the learning process to achieve the optimal joint policy. Theoretically, VDN does not possess the problem discussed above, however, the functional class of the simply additive factorisation is so restrictive [9].

QTRAN. QTRAN gives a sufficient condition for value factorisation that satisfies IGM such that

$$\sum_{i \in \mathcal{N}} Q_i(\mathbf{s}, a_i) - Q^{\pi}(\mathbf{s}, \mathbf{a}) + V^{\pi}(\mathbf{s}) = \begin{cases} 0 & \mathbf{a} = \bar{\mathbf{a}}, \\ \geq 0 & \mathbf{a} \neq \bar{\mathbf{a}}, \end{cases} \tag{18}$$

wherein

$$V^{\pi}(\mathbf{s}) = \max_{\mathbf{a}} Q^{\pi}(\mathbf{s}, \mathbf{a}) - \sum_{i \in \mathcal{N}} Q_i(\mathbf{s}, \bar{a}_i).$$

In Eq.18, $\mathbf{a} = \times_{i \in \mathcal{N}} a_i$; and $\bar{\mathbf{a}} = \times_{i \in \mathcal{N}} \bar{a}_i$ where $\bar{a}_i = \arg\max_{a_i} Q_i(\mathbf{s}, a_i)$ because of IGM. Additionally, Son et al. [10] showed that the above condition also holds for affine transformation on Q_i, $\forall i \in \mathcal{N}$ such that $w_i Q_i + b_i$. For this reason, an additional transformed global Q-value such that $Q^{\pi'}(\mathbf{s}, \mathbf{a}) = \sum_{i \in \mathcal{N}} Q_i(\mathbf{s}, a_i)$ by setting $w_i = 1$ and $\sum_{i \in \mathcal{N}} b_i = 0$ is used to represent the value factorisation. It is forced to fit the above condition with a learned global Q-value $Q^{\pi}(\mathbf{s}, \mathbf{a})$ and $V^{\pi}(\mathbf{s})$. Son et al. [10] argued that finding the factorisation of $Q^{\pi'}(\mathbf{s}, \mathbf{a})$ is equivalent to finding $[Q_i]_{i \in \mathcal{N}}$ to satisfy IGM. Therefore, a value factorisation for obtaining decentralised Q-values that satisfies IGM is found.

D.2 Interpretation of Definitions in Markov Convex Game

D.2.1 Condition of Markov Convex Game

Eq.1 implies a fact existing in most real-life scenarios that a larger coalition results in the greater payoff distributions (see Remark 3) and therefore the greater optimal global value in cooperation, which directly increases the agents' incentives for joining the grand coalition. This can be seen as an insight into the global reward game with value factorisation. This interpretation for the dynamic scenario in this paper is consistent with the static scenario given by [44], also known as the snowball effect.

Remark 3. *Suppose there are two coalitions $\mathcal{T}, \mathcal{S}$ such that $\mathcal{T} \subset \mathcal{S} \subset \mathcal{N}$ and an agent $i \in \mathcal{N} \backslash \mathcal{S}$. For convenience, we denote $\mathcal{C}_1 = \mathcal{T} \cup \{i\}$ and $\mathcal{C}_2 = \mathcal{S}$, and thus $\mathcal{C}_\cap = \mathcal{C}_1 \cap \mathcal{C}_2 = (\mathcal{T} \cup \{i\}) \cap \mathcal{S} = \mathcal{T}$ and $\mathcal{C}_\cup = \mathcal{C}_1 \cup \mathcal{C}_2 = (\mathcal{T} \cup \{i\}) \cup \mathcal{S} = \mathcal{S} \cup \{i\}$. By Eq.1, we can write the following inequalities such that*

$$\begin{aligned} \max_{\pi_{\mathcal{S} \cup \{i\}}} V^{\pi_{\mathcal{S} \cup \{i\}}}(\mathbf{s}) - \max_{\pi_{\mathcal{S}}} V^{\pi_{\mathcal{S}}}(\mathbf{s}) &= \max_{\pi_{\mathcal{C}_\cup}} V^{\pi_{\mathcal{C}_\cup}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}_2}} V^{\pi_{\mathcal{C}_2}}(\mathbf{s}) \\ &\geq \max_{\pi_{\mathcal{C}_1}} V^{\pi_{\mathcal{C}_1}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}_\cap}} V^{\pi_{\mathcal{C}_\cap}}(\mathbf{s}) \\ &= \max_{\pi_{\mathcal{T} \cup \{i\}}} V^{\pi_{\mathcal{T} \cup \{i\}}}(\mathbf{s}) - \max_{\pi_{\mathcal{T}}} V^{\pi_{\mathcal{T}}}(\mathbf{s}). \end{aligned} \tag{19}$$

It is intuitive to see that each agent can gain more payoffs if the size of the coalition grows.

D.2.2 Insight into Markov Core

In Eq.2, $\left(\max_{\pi_i} x_i(\mathbf{s})\right)_{i\in\mathcal{N}}$ indicates the payoff distribution scheme for the grand coalition. $\max_{\pi_\mathcal{C}} x(\mathbf{s}|\mathcal{C}) = \sum_{i\in\mathcal{C}} \max_{\pi_i} x_i(\mathbf{s})$ indicates the sum of payoff distributions (for the grand coalition) of the agents who is under evaluation within coalition $\mathcal{C}$. By Remark 4 and 5, it is obvious that Eq.2 indicates that the optimal global value obtained by the payoff distribution scheme in the Markov core (under the grand coalition) is no less than that they can achieve with other coalition structures, which is called the maximal social welfare in the prior work [13]. It can be regarded as an intuitive interpretation of Markov core (under the grand coalition).

Remark 4. *Suppose that a coalition structure is written as $\mathcal{CS} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n\}$, where $\bigcup_{k=1}^{n} \mathcal{C}_k = \mathcal{N}$ and each $\mathcal{C}_k$ is mutually exclusive (i.e., $\mathcal{C}_m \bigcap \mathcal{C}_n = \varnothing$, if $m \neq n$), the optimal global value with respect to $\mathcal{CS}$ is represented as $\max_\pi V^\pi(\mathbf{s}) = \sum_{k=1}^{n} \max_{\pi_{\mathcal{C}_k}} V^{\pi_{\mathcal{C}_k}}(\boldsymbol{s})$.*

Remark 5. *Suppose that the condition of Markov core holds for the grand coalition (i.e., $\mathcal{N}$) with some payoff distribution scheme $\left(\max_{\pi_i} x_i(\mathbf{s})\right)_{i\in\mathcal{N}}$. For an arbitrary coalition structure $\mathcal{CS} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n\}$ other than $\{\mathcal{N}\}$, where $\bigcup_{k=1}^{n} \mathcal{C}_k = \mathcal{N}$ and each $\mathcal{C}_k$ is mutually exclusive, we can write down the equation such that*

$$\max_{\pi_{\mathcal{C}_k}} x(\mathbf{s}|\mathcal{C}_k) \geq \max_{\pi_{\mathcal{C}_k}} V^{\pi_{\mathcal{C}_k}}(\mathbf{s}), \quad \forall \mathcal{C}_k \in \mathcal{CS}. \tag{20}$$

If we sum up Eq.20 for all coalitions in $\mathcal{CS}$, we can get the following equation such that

$$\sum_{\mathcal{C}_k\in\mathcal{CS}} \max_{\pi_{\mathcal{C}_k}} x(\mathbf{s}|\mathcal{C}_k) \geq \sum_{\mathcal{C}_k\in\mathcal{CS}} \max_{\pi_{\mathcal{C}_k}} V^{\pi_{\mathcal{C}_k}}. \tag{21}$$

Recall that $\max_{\pi_{\mathcal{C}_k}} x(\mathbf{s}|\mathcal{C}_k) = \sum_{j\in\mathcal{C}_k} \max_{\pi_i} x_i(\mathbf{s})$. The LHS of Eq.21 can be written as follows:

$$\sum_{\mathcal{C}_k\in\mathcal{CS}} \max_{\pi_{\mathcal{C}_k}} x(\mathbf{s}|\mathcal{C}_k) = \sum_{\mathcal{C}_k\in\mathcal{CS}} \sum_{j\in\mathcal{C}_k} \max_{\pi_j} x_j(\mathbf{s}) = \sum_{j\in\mathcal{N}} \max_{\pi_j} x_j(\mathbf{s}) = \max_\pi \hat{V}^\pi(\mathbf{s}), \tag{22}$$

wherein $\max_\pi \hat{V}^\pi(\mathbf{s})$ is denoted as the optimal global value obtained by the payoff distribution scheme in the Markov core. By the result in Remark 4, the RHS of Eq.21 can be written as follows:

$$\sum_{\mathcal{C}_k\in\mathcal{CS}} \max_{\pi_{\mathcal{C}_k}} V^{\pi_{\mathcal{C}_k}} = \max_\pi V^\pi(\mathbf{s}), \tag{23}$$

where $\max_\pi V^\pi(\mathbf{s})$ is the optimal global value obtained by an arbitrary coalition structure other than $\{\mathcal{N}\}$. By inserting Eq.22 and 23 into Eq.21, we can get that

$$\max_\pi \hat{V}^\pi(\mathbf{s}) \geq \max_\pi V^\pi(\mathbf{s}).$$

Therefore, we have shown that the solution in the Markov core under the grand coalition is equivalent to the optimal global value.

E Complete Mathematical Proofs

E.1 Assumptions

Assumption 1. *In this paper, we consider a finite Markov convex game, wherein both the state space and the joint action space are finite.*

Assumption 2. *For the ease of analysis, in this paper we assume that each agent's policy will not be affected by the coalition formation. In other words, each agent's policy is regarded as its inherent feature, invariant throughout the interaction with other agents (e.g. joining a coalition).*

Assumption 3. *Any coalition policy can be factorised to a permutation of decentralised (i.e. disjoint) policies, i.e., $\pi_\mathcal{C} = \times_{i\in\mathcal{C}} \pi_i$, where π_i is agent i's policy. Each $\pi_\mathcal{C}$ uniquely corresponds to a $V^{\pi_\mathcal{C}}(\mathbf{s})$ as a characteristic function (i.e. a set-valued function).*

Assumption 4. *If an agent i is a dummy for an arbitrary state $\mathbf{s} \in \mathcal{S}$, it will not provide any contribution to any coalition $\mathcal{C}_i \subseteq \mathcal{N}\backslash\{i\}$ such that $V^{\pi_\mathcal{C}}(\mathbf{s}) = V^{\pi_{\mathcal{C}\cup\{i\}}}(\mathbf{s})$. Additionally, no members in coalition $\mathcal{C}_i$ will react in different manners after agent i joins.*

Assumption 5. *If agents i and j are symmetric for an arbitrary state $\mathbf{s} \in \mathcal{S}$, $V^{\pi_{\mathcal{C}\cup\{i\}}}(\mathbf{s}) = V^{\pi_{\mathcal{C}\cup\{j\}}}(\mathbf{s})$ to any coalitions $\mathcal{C} \subseteq \mathcal{N}\backslash\{i, j\}$. Literally, the contributions of i and j are equal to any coalition $\mathcal{C}$.*

Assumption 6. *For any agent $i \in \mathcal{N}$ and any $\mathbf{s} \in \mathcal{S}$, its optimal Markov Shapley value denoted as $\max_{\pi_i} V_i^{\phi}(\mathbf{s})$ satisfies the following equation such that*

$$\max_{\pi_i} V_i^{\phi}(\mathbf{s}) = \sum_{\mathcal{C}_i \subseteq \mathcal{N}\backslash\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i),$$

where π_i is agent i's policy.

Assumption 1 is the common assumption in the Markov decision process for the ease of analysis. Assumption 2 is a technical assumption for the ease of analysis. Assumption 3 is natural to hold given the chain rule in probability theory, the independence of each agent's policy and the definition of value function in reinforcement learning. Assumption 4 and 5 directly inherit the definitions from cooperative game theory [14]. Assumption 6 inherits the definition from Shapley value [19] with extra consideration of agent i's policy, an underlying condition of which is that the maximizer (i.e., π_i) of each $\Phi_i(\mathbf{s} \mid \mathcal{C}_i) \in \{\Phi_i(\mathbf{s}|\mathcal{C}_i)|\mathcal{C}_i \subseteq \mathcal{N}\backslash\{i\}\}$ needs to be identical, for any $\mathbf{s} \in \mathcal{S}$. In other words, it implies that different permutations correspond to different long-term rewards probably encoding some unexpected events (i.e., each permutation maps to a marginal contribution of agent i), but with the same optimal policy as solutions, which is a sufficient condition for Assumption 2. Thereby, learning through Markov Shapley value is primarily for fair credit assignments, with no changes to each agent's optimal policy. We would argue for the existence of this condition by Example 1.

Example 1. *Suppose that there are two agents in total (i.e., $|\mathcal{N}| = 2$), and we consider an arbitrary agent i belonging to $\mathcal{N}$ whose action set is defined as $\mathcal{A}_i = \{0, 0.15, 0.25\}$. Therefore, there are only two intermediate coalitions for agent i to join and therefore two marginal contributions. To ease life, we only discuss a two-stage scenario and the result can be naturally extended to long-horizon scenarios. Agent i's policy can be expressed as a sequence of actions such that $\pi_i = \langle a_i^0, a_i^1 \rangle$. The set of marginal contributions of agent i is supposed to be $\{\Phi_i(\mathbf{s}|\{-i\}) := -(a_i^0 + a_i^1 - 0.5)^2 + 1 + ||\mathbf{s}||_2, \Phi_i(\mathbf{s}|\varnothing) := \sin(a_i^0 + a_i^1) + ||\mathbf{s}||_2\}$. Since $V_i^{\phi}(\mathbf{s}) = \frac{1}{2}(\Phi_i(\mathbf{s}|\{-i\}) + \Phi_i(\mathbf{s}|\varnothing))$, it is easy to observe that Assumption 6 holds.*

E.2 Mathematical Proofs of The Marginal Contribution

Proposition 4. *$\forall \mathcal{C}_i \subseteq \mathcal{N}$ and $\forall \mathbf{s} \in \mathcal{S}$, Eq.1 is satisfied if and only if $\max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) \geq 0$.*

Proof. $\forall \mathcal{C}_i \subseteq \mathcal{N}$ and $\forall \mathbf{s} \in \mathcal{S}$, given that Eq.1 is satisfied, with the fact that $\mathcal{C}_i \cap \{i\} = \varnothing$ we can get the equation such that

$$\max_{\pi_{\mathcal{C}_i \cup \{i\}}} V^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}) \geq \max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s}) + \max_{\pi_i} V^{\pi_i}(\mathbf{s}). \tag{24}$$

Since $\max_{\pi_i} V^{\pi_i}(\mathbf{s}) \geq 0$ by the definition in Markov convex game, we can easily get the equation such that

$$\max_{\pi_{\mathcal{C}_i \cup \{i\}}} V^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s}) \geq 0. \tag{25}$$

Therefore, we can get the equation such that

$$\max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) \geq 0. \tag{26}$$

With the same conditions, the reverse direction of proof apparently holds by going through from Eq.26 to 24. By Definition 2, Eq.26 determines the range of Markov Shapley value, which is consistent with the range of the coalition value defined in Section 2. □

Proposition 5. *In Markov convex game with the grand coalition, marginal contribution satisfies the efficiency property: $\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i)$.*

Proof. For any $\mathcal{C}_i \subseteq \mathcal{N}\backslash\{i\}$ and $i \in \mathcal{N}$, according to Eq.3 we can get the equation such that

$$\max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) = \max_{\pi_{\mathcal{C}_i \cup \{i\}}} V^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s}), \tag{27}$$

where $\max_{\pi_{\mathcal{C}_i \cup \{i\}}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s}) = \max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s})$, since the decision of agent i will not affect the value of $\mathcal{C}_i$ (i.e., the coalition excluding agent i). Given the definition that $V^{\pi_\varnothing}(\mathbf{s}) = 0$ and the result from Eq.27, by Assumption 3 we can get the equations such that

$$\begin{aligned}
&\max_{\pi} V^{\pi}(\mathbf{s}) \\
&= \max_{\pi_{\{j_1\}}} V^{\pi_{\{j_1\}}}(\mathbf{s}) - \max_{\pi_\varnothing} V^{\pi_\varnothing}(\mathbf{s}) \\
&+ \max_{\pi_{\{j_1,j_2\}}} V^{\pi_{\{j_1\}}}(\mathbf{s}) - \max_{\pi_{\{j_1\}}} V^{\pi_{\{j_1\}}}(\mathbf{s}) \\
&+ \qquad\qquad \vdots \\
&+ \max_{\pi} V^{\pi}(\mathbf{s}) - \max_{\pi_{\mathcal{N}\setminus\{j_n\}}} V^{\pi_{\mathcal{N}\setminus\{j_n\}}}(\mathbf{s}) \\
&= \sum_{i\in\mathcal{N}} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i).
\end{aligned} \tag{28}$$

□

Lemma 1. *The optimal marginal contribution is a solution in the Markov core under Markov convex game with the grand coalition.*

Proof. The complete proof is as follows.

Firstly, if we would like to prove that the optimal marginal contribution is a payoff distribution scheme in the Markov core (with the grand coalition), we just need to prove that for any intermediate coalition $\mathcal{C} \subseteq \mathcal{N}$, the following condition is satisfied such that

$$\max_{\pi_\mathcal{C}} \Phi(\mathbf{s}|\mathcal{C}) \geq \max_{\pi_\mathcal{C}} V^{\pi_\mathcal{C}}(\mathbf{s}), \ \forall \mathbf{s} \in \mathcal{S}, \tag{29}$$

where $\max_{\pi_\mathcal{C}} \Phi(\mathbf{s}|\mathcal{C}) = \sum_{i\in\mathcal{C}} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i)$.

Suppose for the sake of contradiction that we have $\max_{\pi_\mathcal{C}} \Phi(\mathbf{s}|\mathcal{C}) < \max_{\pi_\mathcal{C}} V^{\pi_\mathcal{C}}(\mathbf{s})$ for some $\mathbf{s} \in \mathcal{S}$ and some coalition $\mathcal{C} = \{j_1, j_2, ..., j_{|\mathcal{C}|}\} \subseteq \mathcal{N}$, where $j_n \in \mathcal{C}$ and $n \in \{1, 2, ..., |\mathcal{C}|\}$. We can assume without the loss of generality that the coalition $\mathcal{C}$ is generated by the permutation $\langle j_1, j_2, ..., j_{|\mathcal{C}|}\rangle$, i.e., the agents joins in $\mathcal{C}$ following the order $j_1, j_2, ..., j_{|\mathcal{C}|}$. Now, for each $n \in \{1, 2, ..., |\mathcal{C}|\}$, we have $\{j_1, j_2, ..., j_{n-1}\} \subseteq \{1, 2, ..., j_n - 1\}$. Following Eq.1, we can write out the inequality as follows:

$$\begin{aligned}
\max_{\pi_{\mathcal{C}^n_\cup}} V^{\pi_{\mathcal{C}^n_\cup}}(\mathbf{s}) + \max_{\pi_{\mathcal{C}^n_\cap}} V^{\pi_{\mathcal{C}^n_\cap}}(\mathbf{s}) &\geq \max_{\pi_{\mathcal{C}^n_m}} V^{\pi_{\mathcal{C}^n_m}}(\mathbf{s}) + \max_{\pi_{\mathcal{C}^n_k}} V^{\pi_{\mathcal{C}^n_k}}(\mathbf{s}), \\
\mathcal{C}^n_k &= \{1, 2, ..., j_n - 1\}, \\
\mathcal{C}^n_m &= \{j_1, j_2, ..., j_n\}, \\
\mathcal{C}^n_\cap = \mathcal{C}^n_m \cap \mathcal{C}^n_k &= \{j_1, j_2, ..., j_{n-1}\}, \\
\mathcal{C}^n_\cup = \mathcal{C}^n_m \cup \mathcal{C}^n_k &= \{1, 2, ..., j_n\}.
\end{aligned} \tag{30}$$

Next, we rearrange Eq.30 and the following inequality is obtained such that

$$\max_{\pi_{\mathcal{C}^n_\cup}} V^{\pi_{\mathcal{C}^n_\cup}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}^n_k}} V^{\pi_{\mathcal{C}^n_k}}(\mathbf{s}) \geq \max_{\pi_{\mathcal{C}^n_m}} V^{\pi_{\mathcal{C}^n_m}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}^n_\cap}} V^{\pi_{\mathcal{C}^n_\cap}}(\mathbf{s}), \tag{31}$$

Since we can express $\max_{\pi_\mathcal{C}} V^{\pi_\mathcal{C}}(\mathbf{s})$ as follows:

$$\begin{aligned}
\max_{\pi_\mathcal{C}} V^{\pi_\mathcal{C}}(\mathbf{s}) &= \max_{\pi_{j_1}} V^{\pi_{j_1}}(\mathbf{s}) - \max_{\pi_\varnothing} V^{\pi_\varnothing}(\mathbf{s}) \\
&+ \max_{\pi_{\{j_1,j_2\}}} V^{\pi_{\{j_1,j_2\}}}(\mathbf{s}) - \max_{\pi_{j_1}} V^{\pi_{j_1}}(\mathbf{s}) \\
&+ \qquad\qquad \vdots \\
&+ \max_{\pi_\mathcal{C}} V^{\pi_\mathcal{C}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}\setminus\{j_n\}}} V^{\pi_{\mathcal{C}\setminus\{j_n\}}}(\mathbf{s}).
\end{aligned} \tag{32}$$

By Definition 1 we can obviously get the following equations such that

$$\Phi_i(\mathbf{s}|\mathcal{C}_i) = \Phi_i(\mathbf{s}|\mathcal{C}^n_k) = \max_{\pi_{\mathcal{C}^n_k}} V^{\pi_{\mathcal{C}^n_\cup}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}^n_k}} V^{\pi_{\mathcal{C}^n_k}}(\mathbf{s}). \tag{33}$$

By taking the maximum operator over π_i to Eq.33, we can get that

$$\max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) = \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_k^n) = \max_{\pi_{\mathcal{C}_\cup^n}} V^{\pi_{\mathcal{C}_\cup^n}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}_k^n}} V^{\pi_{\mathcal{C}_k^n}}(\mathbf{s}). \tag{34}$$

By adding up these inequalities in Eq.31 for all $\mathcal{C} \subseteq \mathcal{N}$ and inserting the results from Eq.32 and 34, we can directly obtain a new inequality such that

$$\sum_{i \in \mathcal{C}} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) = \max_{\pi_\mathcal{C}} \Phi(\mathbf{s}|\mathcal{C}) \geq \max_{\pi_\mathcal{C}} V^{\pi_\mathcal{C}}(\mathbf{s}). \tag{35}$$

It is obvious that Eq.35 contradicts the suppose, so we have showed that Eq.29 always holds for any coalition $\mathcal{C} \subseteq \mathcal{N}$. For this reason, we can get the conclusion that marginal contribution is a solution in Markov core of Markov convex game with the grand coalition. □

E.3 Mathematical Proofs of The Markov Shapley Value

Proposition 1. *Agent i's action marginal contribution can be derived as follows:*

$$\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}). \tag{36}$$

Proof. The complete proof is as follows.

We now rewrite $\max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s})$ as follows:

$$\begin{aligned}
\max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}) &= \max_{\pi_{\mathcal{C}_i}} \sum_{\mathbf{a}_{\mathcal{C}_i \cup \{i\}}} \pi_{\mathcal{C}_i \cup \{i\}}(\mathbf{a}_{\mathcal{C}_i \cup \{i\}}|\mathbf{s})\, Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \\
&\quad \big(\text{Since } \pi_{\mathcal{C}_i \cup \{i\}} \text{ is a deterministic joint policy, we can have the following equation.}\big) \\
&= \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \\
&\quad \big(\text{ We write } \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \text{ as } Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \big) \\
&= \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}).
\end{aligned} \tag{37}$$

Similarly, we rewrite $\max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s})$ as follows:

$$\max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s}) = \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) = \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}). \tag{38}$$

Since $\max_{\pi_{\mathcal{C}_i}} V^{\pi_{\mathcal{C}_i}}(\mathbf{s})$ is irrelevant to a_i, by Eq.37 and 38 we can get that

$$\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}). \tag{39}$$

By Eq.39, we can also get Agent i's optimal action marginal contribution such that

$$\begin{aligned}
\Phi^*_i(\mathbf{s}, a_i|\mathcal{C}_i) &= \max_{\pi_i} \Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) \\
&= \max_{\pi_i} \left\{ \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \right\} \\
&= \max_{\pi_i} \left\{ \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \right\} \\
&= \max_{\pi_i} \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \\
&= \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i \cup \{i\}}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \\
&= \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}).
\end{aligned} \tag{40}$$

The proof is completed. □

Proposition 2. *Markov Shapley value possesses properties as follows: (i) identifiability of dummy agents:* $V_i^{\phi}(\mathbf{s}) = 0$*; (ii) efficiency:* $\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} V_i^{\phi}(\mathbf{s})$*; (iii) reflecting the contribution; and (iv) symmetry.*

Proof. The complete proof is as follows.

The marginal contribution is an implementation reflecting an agent's contribution and Markov Shapley value is defined as the weighted average of all marginal contributions. Therefore, this definition can still reflect an agent's contribution to the grand coalition by considering all permutations of agents to form the grand coalition and (iii) holds. We will next prove the (i), followed by (ii) and (iv). For any agent $i \in \mathcal{N}$ and any state $\mathbf{s} \in \mathcal{S}$, its Markov Shapley value denoted as $V_i^{\phi}(\mathbf{s})$.

Proof of (i): Let us define $\Pi(\mathcal{N})$ as the set of all permutations of agents. Suppose that an arbitrary agent i is a dummy agent for an arbitrary state $\mathbf{s} \in \mathcal{S}$. For any permutation $m \in \Pi(\mathcal{N})$ of agents to form the grand coalition, by Assumption 4 we have $\max_{\pi_{\mathcal{C}_i^m}} V^{\pi_{\mathcal{C}_i^m}}(\mathbf{s}) = \max_{\pi_{\mathcal{C}_i^m}} V^{\pi_{\mathcal{C}_i^m \cup \{i\}}}(\mathbf{s})$, thereby $\Phi_i(\mathbf{s}|\mathcal{C}_i^m) = 0$, where $\mathcal{C}_i^m$ denotes the intermediate coalition generated from permutation m that agent i would join. Also, the above analysis is valid for all permutations of agents to form the grand coalition. By Definition 2, it is not difficult to see that the dummy agent's Markov Shapley value will be 0 such that $V_i^{\phi}(\mathbf{s}) = 0$. The proof of (i) completes.

Proof of (ii): The objective is proving that Markov Shapley value satisfies the following equation such that

$$\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} V_i^{\phi}(\mathbf{s}), \quad \forall \mathbf{s} \in \mathcal{S}.$$

By the result from Proposition 5 and Assumption 3, for an arbitrary permutation $m \in \Pi(\mathcal{N})$ we can get the equation such that

$$\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i^m), \quad \forall \mathbf{s} \in \mathcal{S},$$

where $\mathcal{C}_i^m$ denotes the intermediate coalition generated from permutation m that agent i would join and $\Phi_i(\mathbf{s}|\mathcal{C}_i^m)$ is the corresponding marginal contribution. If we consider all possible permutations of agents to form the grand coalition and add all these inequalities, we can get the following equation such that

$$\sum_{m \in \Pi(\mathcal{N})} \max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{m \in \Pi(\mathcal{N})} \sum_{i \in \mathcal{N}} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i^m), \quad \forall \mathbf{s} \in \mathcal{S}.$$

By dividing $|\mathcal{N}|!$ on the both sides, we can get that

$$\frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi} V^{\pi}(\mathbf{s}) = \frac{1}{|\mathcal{N}|!} \sum_{i \in \mathcal{N}} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i^m), \quad \forall \mathbf{s} \in \mathcal{S}. \tag{41}$$

Next, to ease life we start from the LHS of Eq.41. We directly get the following equation such that

$$\frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi} V^{\pi}(\mathbf{s}) = \frac{1}{|\mathcal{N}|!} \cdot |\mathcal{N}|! \cdot \max_{\pi} V^{\pi}(\mathbf{s}) = \max_{\pi} V^{\pi}(\mathbf{s}). \tag{42}$$

Now, we start processing the RHS of Eq.41. By rearranging it, we can get the equations such that

$$\begin{aligned}
\frac{1}{|\mathcal{N}|!} \sum_{i \in \mathcal{N}} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) &= \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) \\
&\quad \text{(The identical } \mathcal{C}_i^m \text{ in different permutations is written as } \mathcal{C}_i \\
&\quad \text{and we can rearrange the equation as follows.)} \\
&= \sum_{i \in \mathcal{C}} \frac{1}{|\mathcal{N}|!} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}} |\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)! \cdot \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) \\
&= \sum_{i \in \mathcal{N}} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i).
\end{aligned} \tag{43}$$

By Assumption 6, we can get the following equations such that

$$\sum_{i\in\mathcal{N}}\sum_{\mathcal{C}_i\subseteq\mathcal{N}\backslash\{i\}}\frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}\cdot\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i)=\sum_{i\in\mathcal{N}}\max_{\pi_i}V_i^{\phi}(\mathbf{s}) \tag{44}$$

Inserting the results from Eq.42 and 44 to Eq.41, we can get the equation such that

$$\max_{\pi}V^{\pi}(\mathbf{s})=\sum_{i\in\mathcal{N}}\max_{\pi_i}V_i^{\phi}(\mathbf{s}),\quad\forall\mathbf{s}\in\mathcal{S}.$$

Therefore, the proof for (ii) completes.

Proof of (iv): We would like to prove that if two agents are symmetric for an arbitrary state $\mathbf{s}\in\mathcal{S}$, then their optimal Markov Shapley values should be equal. As Assumption 5 illustrates, suppose that agents i and j are symmetric for an arbitrary state $\mathbf{s}\in\mathcal{S}$, $V^{\pi_{\mathcal{C}\cup\{i\}}}(\mathbf{s})=V^{\pi_{\mathcal{C}\cup\{j\}}}(\mathbf{s})$ for any coalitions $\mathcal{C}\subseteq\mathcal{N}\backslash\{i,j\}$. Given an arbitrary permutation $m\in\Pi(\mathcal{N})$, let m' denote the permutation obtained by exchanging i and j such that $\mathcal{C}_i^m=\mathcal{C}_j^{m'}$, $\mathcal{C}_i^{m'}=\mathcal{C}_j^m$ and $\mathcal{C}_l^{m'}=\mathcal{C}_l^m,\forall l\neq i,j$. Next, we aim to prove that $\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i^m)=\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j^{m'})$, for the state $\mathbf{s}$.

We first suppose that i precedes j in m. Then we have $\mathcal{C}_i^m=\mathcal{C}_j^{m'}$. Setting $\mathcal{C}=\mathcal{C}_i^m=\mathcal{C}_j^{m'}$, for the state $\mathbf{s}$ we can obtain that

$$\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i^m)=\max_{\pi_{\mathcal{C}\cup\{i\}}}V^{\pi_{\mathcal{C}\cup\{i\}}}(\mathbf{s})-\max_{\pi_{\mathcal{C}}}V^{\pi_{\mathcal{C}}}(\mathbf{s}),$$
$$\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j^{m'})=\max_{\pi_{\mathcal{C}\cup\{j\}}}V^{\pi_{\mathcal{C}\cup\{j\}}}(\mathbf{s})-\max_{\pi_{\mathcal{C}}}V^{\pi_{\mathcal{C}}}(\mathbf{s}).$$

By symmetry, we have $V^{\pi_{\mathcal{C}\cup\{i\}}}(\mathbf{s})=V^{\pi_{\mathcal{C}\cup\{j\}}}(\mathbf{s})$, which directly implies that $\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i^m)=\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j^{m'})$.

Second, we suppose that j precedes i in m. Setting $\mathcal{C}=\mathcal{C}_i^m\backslash\{j\}$, for the state $\mathbf{s}$ we have

$$\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i^m)=\max_{\pi_{\mathcal{C}\cup\{j\}\cup\{i\}}}V^{\pi_{\mathcal{C}\cup\{j\}\cup\{i\}}}(\mathbf{s})-\max_{\pi_{\mathcal{C}\cup\{j\}}}V^{\pi_{\mathcal{C}\cup\{j\}}}(\mathbf{s}),$$
$$\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j^{m'})=\max_{\pi_{\mathcal{C}\cup\{j\}\cup\{i\}}}V^{\pi_{\mathcal{C}\cup\{j\}\cup\{i\}}}(\mathbf{s})-\max_{\pi_{\mathcal{C}\cup\{i\}}}V^{\pi_{\mathcal{C}\cup\{i\}}}(\mathbf{s}).$$

Since $\mathcal{C}\subseteq\mathcal{N}\backslash\{i,j\}$, by symmetry we have $V^{\pi_{\mathcal{C}\cup\{j\}}}(\mathbf{s})=V^{\pi_{\mathcal{C}\cup\{i\}}}(\mathbf{s})$ and thus $\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i^m)=\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j^{m'})$. Therefore, we have proved that $\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i^m)=\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j^{m'})$ for any $m\in\Pi(\mathcal{N})$. It is not difficult to observe that $m\mapsto m'$ is a one-to-one mapping, so $\Pi(\mathcal{N})=\{m'\mid m\in\Pi(\mathcal{N})\}$.

By Assumption 6, for an arbitrary state $\mathbf{s}\in\mathcal{S}$ wherein agents are symmetric, we can directly have

$$\begin{aligned}\max_{\pi_i}V_i^{\phi}(\mathbf{s})&=\sum_{\mathcal{C}_i\subseteq\mathcal{N}\backslash\{i\}}\frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}\cdot\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i)\\&=\frac{1}{|\mathcal{N}|!}\sum_{m\in\Pi(\mathcal{N})}\max_{\pi_i}\Phi_i(\mathbf{s}|\mathcal{C}_i^m)\\&=\frac{1}{|\mathcal{N}|!}\sum_{m'\in\Pi(\mathcal{N})}\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j^{m'})\\&=\sum_{\mathcal{C}_j\subseteq\mathcal{N}\backslash\{j\}}\frac{|\mathcal{C}_j|!(|\mathcal{N}|-|\mathcal{C}_j|-1)!}{|\mathcal{N}|!}\cdot\max_{\pi_j}\Phi_j(\mathbf{s}|\mathcal{C}_j)\\&=\max_{\pi_j}V_j^{\phi}(\mathbf{s}).\end{aligned}$$

The proof of (iv) completes. □

E.4 Mathematical Proofs and Derivations for Shapley Q-Learning

E.4.1 Derivation of Shapley-Bellman optimality equation.

First, according to Bellman's principle of optimality [15, 16] we can write out Bellman optimality equation for the optimal global Q-value such that

$$Q^{\pi^*}(\mathbf{s}, \mathbf{a}) = \sum_{\mathbf{s}'} Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \Big[R + \gamma \max_{\mathbf{a}} Q^{\pi^*}(\mathbf{s}', \mathbf{a}) \Big]. \tag{45}$$

For convenience, we only consider the finite state space and action space here. By the efficiency property (i.e. (ii) in Proposition 2), we can get the approximation of the optimal global Q-value w.r.t. optimal actions such that

$$\max_{\mathbf{a}} Q^{\pi^*}(\mathbf{s}', \mathbf{a}) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}', a_i). \tag{46}$$

Suppose that for all $\mathbf{s} \in \mathcal{S}$ and $a_i \in \mathcal{A}_i$, for each agent i there exists bounded $w_i(\mathbf{s}, a_i) > 0$ and $b_i(\mathbf{s}) \geq 0$ that can project $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ onto the space of $Q_i^{\phi^*}(\mathbf{s}, a_i)$ such that

$$Q_i^{\phi^*}(\mathbf{s}, a_i) = w_i(\mathbf{s}, a_i)\, Q^{\pi^*}(\mathbf{s}, \mathbf{a}) - b_i(\mathbf{s}). \tag{47}$$

If we denote $\mathbf{w}(\mathbf{s}, \mathbf{a}) = [w_i(\mathbf{s}, a_i)]^\top \in \mathbb{R}_{>0}^{|\mathcal{N}|}$, $\mathbf{b}(\mathbf{s}) = [b_i(\mathbf{s})]^\top \in \mathbb{R}_{\geq 0}^{|\mathcal{N}|}$ and $\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = [Q_i^{\phi^*}(\mathbf{s}, a_i)]^\top \in \mathbb{R}_{\geq 0}^{|\mathcal{N}|}$, given Eq.47 we can write that

$$\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = \mathbf{w}(\mathbf{s}, \mathbf{a})\, Q^{\pi^*}(\mathbf{s}, \mathbf{a}) - \mathbf{b}(\mathbf{s}). \tag{48}$$

Besides, we suppose that $\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}) = 0$.

Combined with Eq.46 and 48, we can rewrite Eq.45 to the equation as follows:

$$\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = \mathbf{w}(\mathbf{s}, \mathbf{a}) \sum_{\mathbf{s}'} Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \Bigg[R \; + \; \gamma \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}', a_i) \Bigg] - \mathbf{b}(\mathbf{s}). \tag{49}$$

From Eq.47, we know that $w_i(\mathbf{s}, a_i) > 0$. Therefore, we can rewrite Eq.47 to the following equation such that

$$w_i(\mathbf{s}, a_i)^{-1} \; \Big(Q_i^{\phi^*}(\mathbf{s}, a_i) + b_i(\mathbf{s}) \Big) = Q^{\pi^*}(\mathbf{s}, \mathbf{a}). \tag{50}$$

If we sum up Eq.50 for all agents, we can obtain that

$$\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} \; \Big(Q_i^{\phi^*}(\mathbf{s}, a_i) + b_i(\mathbf{s}) \Big) = |\mathcal{N}|\, Q^{\pi^*}(\mathbf{s}, \mathbf{a}). \tag{51}$$

Since $\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}) = 0$, we can get the following equation such that

$$\sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)}\, Q_i^{\phi^*}(\mathbf{s}, a_i) = \; Q^{\pi^*}(\mathbf{s}, \mathbf{a}). \tag{52}$$

Inserting Eq.46 into Eq.52, we can get the following equation such that

$$\max_{\mathbf{a}} \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)}\, Q_i^{\phi^*}(\mathbf{s}, a_i) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i). \tag{53}$$

Since $\mathbf{a} = \times_{i \in \mathcal{N}} a_i$, we can get that

$$\sum_{i \in \mathcal{N}} \max_{a_i} \frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)}\, Q_i^{\phi^*}(\mathbf{s}, a_i) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i). \tag{54}$$

It is apparent that $\forall \mathbf{s} \in \mathcal{S}$ and $a_i^* = \arg\max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i)$, we have a solution $w_i(\mathbf{s}, a_i^*) = 1/|\mathcal{N}|$. [7]

[7] Note that it exists other solutions rather than the one that we deduce between $\max_{a_i} \frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)}\, Q_i^{\phi^*}(\mathbf{s}, a_i)$ and $\max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i)$. Nevertheless, the result obtained in this paper is the one that exactly matches and explains the finding in the previous works [20]. As for the reason why the solution is the most likely to be achieved in empirical results is deserved to be studied in the future work.

E.4.2 Proof of Theorem 1

Lemma 2 (Dales et al. [45]). *A set of real matrices $\mathcal{M}$ with a sub-multiplicative norm is a Banach Algebra and a non-empty complete metric space where the metric is induced by the sub-multiplicative norm. A sub-multiplicative norm $\|\cdot\|$ is a norm satisfying the following inequality such that*

$$\forall \mathbf{A}, \mathbf{B} \in \mathcal{M} : \|\mathbf{AB}\| \leq \|\mathbf{A}\| \, \|\mathbf{B}\|.$$

Lemma 3. *For a set of real matrices $\mathcal{M}$, given an arbitrary matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$, $\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{1 \leq i \leq m} |a_{ij}|$ is a sub-multiplicative norm.*

Proof. The complete proof is as follows.

First, we select two arbitrary matrices belonging to $\mathcal{M}$, i.e. $\mathbf{A} = [a_{ik}] \in \mathbb{R}^{m \times r}$ and $\mathbf{B} = [b_{kj}] \in \mathbb{R}^{r \times n}$. Then, we start proving that $\|\cdot\|_1$ is a sub-multiplicative norm as follows:

$$\begin{aligned}
\|\mathbf{AB}\|_1 &= \left\| \left[\sum_{1 \leq k \leq r} a_{ik} b_{kj} \right] \right\|_1 \\
&= \max_{1 \leq j \leq n} \sum_{1 \leq i \leq m} \left| \sum_{1 \leq k \leq r} a_{ik} b_{kj} \right| \\
&\quad \text{(By triangle inequality, we can obtain the following inequality.)} \\
&\leq \max_{1 \leq j \leq n} \sum_{1 \leq i \leq m} \sum_{1 \leq k \leq r} \left| a_{ik} b_{kj} \right| \\
&= \max_{1 \leq j \leq n} \sum_{1 \leq i \leq m} \sum_{1 \leq k \leq r} \left| a_{ik} \right| \left| b_{kj} \right| \\
&= \max_{1 \leq j \leq n} \sum_{1 \leq k \leq r} \left| b_{kj} \right| \sum_{1 \leq i \leq m} \left| a_{ik} \right| \\
&\leq \left\| \mathbf{B} \right\|_1 \max_{1 \leq k \leq r} \sum_{1 \leq i \leq m} \left| a_{ik} \right| \\
&= \left\| \mathbf{B} \right\|_1 \left\| \mathbf{A} \right\|_1 \\
&= \left\| \mathbf{A} \right\|_1 \left\| \mathbf{B} \right\|_1.
\end{aligned}$$

Therefore, we have proven that given an arbitrary real matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$, $\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{1 \leq i \leq m} |a_{ij}|$ is a sub-multiplicative norm. □

Lemma 4. *For all $\mathbf{s} \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$, Shapley-Bellman operator is a contraction mapping in a non-empty complete metric space when $\max_{\mathbf{s}} \left\{ \sum_{i \in \mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \right\} < \frac{1}{\gamma}$.*

Proof. The complete proof is as follows.

To ease life, we firstly define some variables that will be used for proof such that

$$\begin{aligned}
&\mathbf{Q}^{\phi} = \times_{i \in \mathcal{N}} Q_i^{\phi} \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{S}||\mathcal{A}|}, \\
&\mathbf{w} \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{S}||\mathcal{A}|}, \\
&Pr \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}, \\
&\mathbf{1} = [1, 1, ..., 1]^{\top},
\end{aligned}$$

where $\mathcal{A} = \times_{i \in \mathcal{N}} \mathcal{A}_i$. Then, for an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we define the $\|\cdot\|_1$ for the induced matrix norm such that

$$\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{1 \leq i \leq m} |a_{ij}|,$$

where a_{ij} is an arbitrary element in $\mathbf{A}$. By Lemma 3, $\|\cdot\|_1$ defined here is a sub-multiplicative norm. By Lemma 2, the set of real matrices $\mathbb{R}^{|\mathcal{N}| \times |\mathcal{S}||\mathcal{A}|}$ with the norm $\|\cdot\|_1$ is a Banach algebra and a non-empty complete metric space with the metric induced by $\|\cdot\|_1$.

To show that the operator Υ is a contraction mapping in the supremum norm, we just need to show that for any $\mathbf{Q}_1^{\phi} = \times_{i \in \mathcal{N}} \left(Q_i^{\phi} \right)_1 \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{S}||\mathcal{A}|}$ and $\mathbf{Q}_2^{\phi} = \times_{i \in \mathcal{N}} \left(Q_i^{\phi} \right)_2 \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{S}||\mathcal{A}|}$, we have

$\|\Upsilon\mathbf{Q}_1^\phi - \Upsilon\mathbf{Q}_2^\phi\|_1 \le \delta\|\mathbf{Q}_1^\phi - \mathbf{Q}_2^\phi\|_1$, where $\delta \in (0,1)$.

$$
\begin{aligned}
&\|\Upsilon\mathbf{Q}_1^\phi - \Upsilon\mathbf{Q}_2^\phi\|_1 \\
&= \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^\top \Big| \mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a}) \Big[R(\mathbf{s},\mathbf{a}) + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) \Big] - \mathbf{b}(\mathbf{s}) \\
&\quad - \mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a}) \Big[R(\mathbf{s},\mathbf{a}) + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big] + \mathbf{b}(\mathbf{s}) \Big| \\
&= \gamma \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^\top \Big| \mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a}) \Big[\sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big] \Big| \\
&\le \gamma \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^\top \Big| \mathbf{w}(\mathbf{s},\mathbf{a}) \Big| \max_{\mathbf{s},\mathbf{a}} \Big| \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a}) \Big[\sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big] \Big| \\
&\quad \Big(\text{If we write } \delta = \gamma \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^\top \big|\mathbf{w}(\mathbf{s},\mathbf{a})\big|, \text{ we can have the following equation.} \Big) \\
&= \delta \max_{\mathbf{s},\mathbf{a}} \Big| \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a}) \Big[\sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big] \Big| \\
&\le \delta \max_{\mathbf{s},\mathbf{a}} \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a}) \Big| \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big| \\
&= \delta \Big| \sum_{i\in\mathcal{N}} \Big[\max_{a_i} \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \max_{a_i} \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big] \Big| \\
&\quad \text{(By triangle inequality, we can obtain the following inequality.)} \\
&\le \delta \sum_{i\in\mathcal{N}} \Big| \max_{a_i} \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \max_{a_i} \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big| \\
&\le \delta \sum_{i\in\mathcal{N}} \max_{a_i} \Big| \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big| \\
&\quad \text{(Since } \mathbf{a} = \times_{i\in\mathcal{N}} a_i, \text{ we have the following equation.)} \\
&= \delta \max_{\mathbf{a}} \sum_{i\in\mathcal{N}} \Big| \big(Q_i^\phi\big)_1(\mathbf{s}', a_i) - \big(Q_i^\phi\big)_2(\mathbf{s}', a_i) \Big| \\
&\le \delta \max_{\mathbf{z},\mathbf{a}} \sum_{i\in\mathcal{N}} \Big| \big(Q_i^\phi\big)_1(\mathbf{z}, a_i) - \big(Q_i^\phi\big)_2(\mathbf{z}, a_i) \Big| = \delta\|\mathbf{Q}_1^\phi - \mathbf{Q}_2^\phi\|_1.
\end{aligned}
$$

Now, we need to discuss the condition to $\delta \in (0,1)$. Apparently, $\delta > 0$, so we just need to discuss the condition to guarantee that $\delta < 1$. We now have the following discussions such that

$$
\begin{aligned}
&\delta = \gamma \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^\top \big|\mathbf{w}(\mathbf{s},\mathbf{a})\big| < 1 \text{ (Since } w_i(\mathbf{s}, a_i) > 0.) \\
&\Rightarrow \gamma \max_{\mathbf{s},\mathbf{a}} \sum_{i\in\mathcal{N}} w_i(\mathbf{s}, a_i) < 1 \\
&\text{(When } \gamma \neq 0, \text{ we can have the following inequality.)} \\
&\Rightarrow \max_{\mathbf{s},\mathbf{a}} \sum_{i\in\mathcal{N}} w_i(\mathbf{s}, a_i) < \frac{1}{\gamma} \\
&\text{(Since } \mathbf{a} = \times_{i\in\mathcal{N}} a_i, \text{ we have the following equation.)} \\
&\Rightarrow \max_{\mathbf{s}} \Big\{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \Big\} < \frac{1}{\gamma}.
\end{aligned}
$$

Therefore, we show that Shapley-Bellman operator Υ is a contraction mapping in the non-empty complete metric space generated by $\mathbb{R}^{|\mathcal{N}|\times|\mathcal{S}||\mathcal{A}|}$ with the metric induced by $\|\cdot\|_1$, when $\max_{\mathbf{s}} \Big\{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \Big\} < \frac{1}{\gamma}$. Finally, it is apparent that $w_i(\mathbf{s}, a_i) = 1/|\mathcal{N}|$ when $a_i = \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i)$ satisfies the above condition. □

Corollary 1. *According to Banach fixed-point theorem [46], Shapley-Bellman operator admits a unique fixed point. Moreover, starting by an arbitrary start point, the sequence recursively generated by Shapley-Bellman operator can finally converge to that fixed point.*

Proof. Since $\langle \mathbb{R}^{|\mathcal{N}|\times|\mathcal{S}||\mathcal{A}|}, ||\cdot||_1 \rangle$ is a non-empty complete metric space and Shapley-Bellman operator Υ is shown as a contraction mapping in Lemma 4, by Banach fixed-point theorem [46] we can directly conclude that Shapley-Bellman operator Υ admits a unique fixed point. Furthermore, starting by an arbitrary start point, the sequence recursively generated by Shapley-Bellman operator Υ can finally converge to that fixed point. □

Theorem 1. *Shapley-Bellman operator can converge to the optimal Markov Shapley Q-value and the corresponding optimal joint deterministic policy when* $\max_{\mathbf{s}} \left\{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \right\} < \frac{1}{\gamma}$.

Proof. By Corollary 1, we get that Shapley-Bellman operator admits a unique fixed point. Since Shapley-Bellman optimality equation (i.e., Eq.7) is obviously a fixed point for Shapley-Bellman operator, it is not difficult to get the conclusion that the optimal Markov Shapley Q-value is achieved. Since the sum of optimal Markov Shapley Q-values is equal to the optimal global Q-value and the optimal global Q-value corresponds to the optimal joint deterministic policy, we show that the optimal joint deterministic policy is achieved. Besides, it is obvious that Shapley-Bellman optimality equation can be transformed back to the Bellman optimality equation w.r.t. the optimal global Q-value, given the efficiency property of Markov Shapley value. □

E.4.3 Stochastic Approximation of Shapley-Bellman operator

We now derive the stochastic approximation of Shapley-Bellman operator over the value space, i.e. a form of Q-learning derived from Shapley-Bellman operator. By sampling from $Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ via Monte Carlo method, the Q-learning algorithm can be expressed as follows:

$$\mathbf{Q}^{\phi}_{t+1}(\mathbf{s}, \mathbf{a}) \leftarrow \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) + \alpha_t(\mathbf{s}, \mathbf{a}) \big[\mathbf{w}(\mathbf{s}, \mathbf{a}) \big(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} (Q^{\phi}_i)_t(\mathbf{s}', a_i) \big) - \mathbf{b}(\mathbf{s}) - \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) \big]. \quad (55)$$

Lemma 5 (Jaakkola et al. [47])**.** *The random process* $\{\Delta_t\}$ *taking values* $\mathbb{R}^n$ *defined as*

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\Delta_t(x) + \alpha_t(x)F_t(x)$$

converges to 0 w.p.1 under the following assumptions:

- $0 \le \alpha_t \le 1$, $\sum_t \alpha_t(x) = \infty$ *and* $\sum_t \alpha_t^2 \le \infty$*;*
- $||\mathbb{E}[F_t(x)|\mathcal{F}_t]||_W \le \delta ||\Delta_t||_W$*, with* $0 \le \delta < 1$*;*
- ***var***$[F_t(x)|\mathcal{F}_t] \le C(1 + ||\Delta_t||^2_W)$*, for* $C > 0$*.*

Theorem 4. *For a finite Markov convex game, the Q-learning algorithm derived by Shapley-Bellman operator given by the update rule such that*

$$\mathbf{Q}^{\phi}_{t+1}(\mathbf{s}, \mathbf{a}) \leftarrow \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) + \alpha_t(\mathbf{s}, \mathbf{a}) \left[\mathbf{w}(\mathbf{s}, \mathbf{a}) \left(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} (Q^{\phi}_i)_t(\mathbf{s}', a_i) \right) - \mathbf{b}(\mathbf{s}) - \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) \right],$$

converges w.p.1 to the optimal Markov Shapley Q-value if

$$\sum_t \alpha_t(\mathbf{s}, \mathbf{a}) = \infty \qquad \sum_t \alpha_t^2(\mathbf{s}, \mathbf{a}) \le \infty \quad (56)$$

for all $\mathbf{s} \in \mathcal{S}$ *and* $\mathbf{a} \in \mathcal{A}$ *as well as* $\max_{\mathbf{s}} \{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \} < \frac{1}{\gamma}$.

Proof. The proof follows the sketch of proving the convergence of Q-learning given by Melo [48]. First, we rewrite Eq.55 to

$$\mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) = (1 - \alpha_t(\mathbf{s}, \mathbf{a}))\, \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) + \alpha_t(\mathbf{s}, \mathbf{a}) \left[\mathbf{w}(\mathbf{s}, \mathbf{a}) \left(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} (Q^{\phi}_i)_t(\mathbf{s}', a_i) \right) - \mathbf{b}(\mathbf{s}) \right].$$

By subtracting $\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a})$ and letting

$$\Delta_t(\mathbf{s}, \mathbf{a}) = \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) - \mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}),$$

we can transform Eq.55 to

$$\Delta_{t+1}(\mathbf{s},\mathbf{a}) = (1-\alpha_t(\mathbf{s},\mathbf{a}))\Delta_t(\mathbf{s},\mathbf{a}) + \alpha_t(\mathbf{s},\mathbf{a})F_t(\mathbf{s},\mathbf{a}),$$

where

$$F_t(\mathbf{s},\mathbf{a}) = \mathbf{w}(\mathbf{s},\mathbf{a})\left(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\right) - \mathbf{b}(\mathbf{s}) - \mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a}).$$

Since $\mathbf{s}' \in \mathcal{S}$ is a random sample from Markov Chain, so we can get that

$$\begin{aligned}
\mathbb{E}[F_t(\mathbf{s},\mathbf{a})|\mathcal{F}_t] &= \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a})\left[\mathbf{w}(\mathbf{s},\mathbf{a})\left(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\right) - \mathbf{b}(\mathbf{s}) - \mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a})\right] \\
&= \mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s},\mathbf{a})\left(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\right) - \mathbf{b}(\mathbf{s}) - \mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a}) \\
&\quad \left(\text{Since } \max_{\mathbf{s}}\Big\{\sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s},a_i)\Big\} < \frac{1}{\gamma}.\right) \\
&= \Upsilon\mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a}) - \Upsilon\mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a}).
\end{aligned}$$

By the results from Theorem 4, we can get that

$$\|\mathbb{E}[F_t(\mathbf{s},\mathbf{a})|\mathcal{F}_t]\|_1 \le \delta\|\mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a}) - \mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a})\|_1 = \delta\|\Delta_t(\mathbf{s},\mathbf{a})\|_1,$$

where $\delta \in (0,1)$.

Next, we get that

$$\begin{aligned}
\mathbf{var}[F_t(\mathbf{s},\mathbf{a})|\mathcal{F}_t] &= \mathbb{E}\Big[\Big(\mathbf{w}(\mathbf{s},\mathbf{a})\Big(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\Big) - \mathbf{b}(\mathbf{s}) - \mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a}) \\
&\quad - \Upsilon\mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a}) + \mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a})\Big)^2\Big] \\
&= \mathbb{E}\Big[\Big(\mathbf{w}(\mathbf{s},\mathbf{a})\Big(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\Big) - \mathbf{b}(\mathbf{s}) - \Upsilon\mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a})\Big)^2\Big] \\
&= \mathbf{var}\Big[\mathbf{w}(\mathbf{s},\mathbf{a})\Big(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\Big) - \mathbf{b}(\mathbf{s})\Big|\mathcal{F}_t\Big].
\end{aligned}$$

Since R_t, $\mathbf{w}(\mathbf{s},\mathbf{a})$ and $\mathbf{b}(\mathbf{s})$ are bounded, it clearly verifies that

$$\mathbf{var}[F_t(\mathbf{s},\mathbf{a})|\mathcal{F}_t] \le C(1 + \|\Delta_t(\mathbf{s},\mathbf{a})\|_1^2)$$

for some constant C.

Finally, by Lemma 5 it is easy to see that Δ_t converges to 0 w.p.1, i.e., $\mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a})$ converges to $\mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a})$ w.p.1, given the condition in Eq.56. □

E.4.4 Derivation of Shapley Q-Learning

Similar to the operations in Section E.4.3, by stochastic approximation in value space, i.e. sampling $\mathbf{s}'$ from $Pr(\mathbf{s}'|\mathbf{s},\mathbf{a})$ via Monte Carlo method, Shapley-Bellman operator can be expressed as follows:

$$\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = \mathbf{w}(\mathbf{s},\mathbf{a})\left(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}',a_i)\right) - \mathbf{b}(\mathbf{s}), \tag{57}$$

where $\mathbf{w}(\mathbf{s},\mathbf{a}) = [w_i(\mathbf{s},a_i)]^{\top} \in \mathbb{R}_+^{|\mathcal{N}|}$; $\mathbf{b}(\mathbf{s}) = [b_i(\mathbf{s})]^{\top} \in \mathbb{R}_+^{|\mathcal{N}|}$; and $\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = [Q_i^{\phi}(\mathbf{s},a_i)]^{\top} \in \mathbb{R}_+^{|\mathcal{N}|}$. Since $\mathbf{w}(\mathbf{s},\mathbf{a}) = diag\big(\mathbf{w}(\mathbf{s},\mathbf{a})\big)\,\mathbf{1}$ where $diag(\cdot)$ denotes the diagonalization of a vector[8] and $\mathbf{1}$ denotes the vector of ones, Eq.57 can be equivalently represented as

$$\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = diag\big(\mathbf{w}(\mathbf{s},\mathbf{a})\big)\,\mathbf{1}\left(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}',a_i)\right) - \mathbf{b}(\mathbf{s}). \tag{58}$$

[8]It is a square diagonal matrix with the elements of vector v on the main diagonal, and the other entries of the matrix are zeros.

Since $w_i(\mathbf{s}, a_i) > 0, \forall i \in \mathcal{N}$, we can write the following equivalent form to Eq.58 such that

$$diag\big(\mathbf{w}(\mathbf{s},\mathbf{a})\big)^{-1}\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = \mathbf{1}\left(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i)\right) - diag\big(\mathbf{w}(\mathbf{s},\mathbf{a})\big)^{-1}\mathbf{b}(\mathbf{s}). \tag{59}$$

Next, we multiply $\mathbf{1}^{\top}$ on both sides and obtain the following equation such that

$$\sum_{i\in\mathcal{N}} \frac{1}{w_i(\mathbf{s}, a_i)}\, Q_i^{\phi}(\mathbf{s}, a_i) = |\mathcal{N}|\left(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i)\right) - \sum_{i\in\mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}). \tag{60}$$

Since the condition such that $\sum_{i\in\mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}) = 0$, by dividing $|\mathcal{N}|$ on both sides we get that

$$\sum_{i\in\mathcal{N}} \frac{1}{|\mathcal{N}| w_i(\mathbf{s}, a_i)}\, Q_i^{\phi}(s, a_i) = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i). \tag{61}$$

Since $w_i(\mathbf{s}, a_i) = 1/|\mathcal{N}|$ when $a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i)$, by defining $\delta_i(\mathbf{s}, a_i) = \frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)}$ we can get that

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \end{cases} \tag{62}$$

where $\alpha_i(\mathbf{s}, a_i)$ is a variable that expresses $\frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)}$ when $a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i)$ for the ease of implementation.

Substituting Eq.62 into Eq.61, we can get the following equation such that

$$\sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\, Q_i^{\phi}(\mathbf{s}, a_i) = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i). \tag{63}$$

By rearranging Eq.63, we obtain the TD error of Shapley Q-learning (SHAQ) such that

$$\Delta(\mathbf{s}, \mathbf{a}, \mathbf{s}') = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\, Q_i^{\phi}(\mathbf{s}, a_i). \tag{64}$$

Note that the TD error of SHAQ is necessary for the TD error of Eq.55 (i.e. the stochastic learning process that we proved to converge to the optimal Markov Shapley Q-value in Theorem 4). For this reason, the condition $\max_{\mathbf{s}} \left\{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \right\} < \frac{1}{\gamma}$ is necessary to be satisfied so that the convergence to the optimality is possible to hold.

E.5 Mathematical Proofs of Validity and Interpretability

Lemma 6. *Markov core is a convex set.*

Proof. Let $\left(\max_{\pi_i} x_i(\mathbf{s})\right)_{i\in\mathcal{N}}$ and $\left(\max_{\pi_i} y_i(\mathbf{s})\right)_{i\in\mathcal{N}}$ be two vectors in the Markov core and $\alpha \in [0, 1)$ be an arbitrary scalar. To ease life, for any $i \in \mathcal{N}$ we let $\max_{\pi_i} z_i(\mathbf{s}) = \alpha \max_{\pi_i} x_i(\mathbf{s}) + (1 - \alpha) \max_{\pi_i} y_i(\mathbf{s})$. By definition, for any coalition $\mathcal{C} \subseteq \mathcal{N}$ we have

$$\begin{aligned}
\max_{\pi_{\mathcal{C}}} z(\mathbf{s}|\mathcal{C}) &= \sum_{i\in\mathcal{C}} \max_{\pi_i} z_i(\mathbf{s}) \\
&= \sum_{i\in\mathcal{C}} \alpha \max_{\pi_i} x_i(\mathbf{s}) + (1-\alpha) \max_{\pi_i} y_i(\mathbf{s}) \\
&= \alpha \sum_{i\in\mathcal{C}} \max_{\pi_i} x_i(\mathbf{s}) + (1-\alpha) \sum_{i\in\mathcal{C}} \max_{\pi_i} y_i(\mathbf{s}) \\
&\geq \alpha \max_{\pi_{\mathcal{C}}} V^{\pi_{\mathcal{C}}}(\mathbf{s}) + (1-\alpha) \max_{\pi_{\mathcal{C}}} V^{\pi_{\mathcal{C}}}(\mathbf{s}) \\
&= \max_{\pi_{\mathcal{C}}} V^{\pi_{\mathcal{C}}}(\mathbf{s}).
\end{aligned}$$

Therefore, we proved that Markov core is a convex set. □

Theorem 2. *The optimal Markov Shapley value is a solution in the Markov core under Markov convex game (MCG) with the grand coalition.*

Proof. The optimal Markov Shapley value is the affine combination of the optimal marginal contributions. We know that Markov core is a convex set by Lemma 6 and the optimal marginal contribution is in the Markov core by Lemma 1. Since the affine combination of the points in a convex set is still in this convex set, we get that the optimal Markov Shapley value is in the Markov core. □

E.6 Mathematical Derivation for Implementation of Shapley Q-Learning

Proposition 3. *Suppose any action marginal contribution can be factorised to the form such that* $\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = \sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})\ \hat{Q}_i(\mathbf{s}, a_i)$. *With the condition such that*

$$\mathbb{E}_{\mathcal{C}_i\sim Pr(\mathcal{C}_i|\mathcal{N}\setminus\{i\})}[\sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})] = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i), \\ K \in (0,1) & a_i \neq \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i), \end{cases}$$

we have

$$\begin{cases} Q_i^\phi(\mathbf{s}, a_i) = \hat{Q}_i(\mathbf{s}, a_i) & a_i = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i)\ Q_i^\phi(\mathbf{s}, a_i) = \hat{\alpha}_i(\mathbf{s}, a_i)\ \hat{Q}_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \end{cases}$$

where $\hat{\alpha}_i(\mathbf{s}, a_i) = \mathbb{E}_{\mathcal{C}_i\sim Pr(\mathcal{C}_i|\mathcal{N}\setminus\{i\})}[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})]$ *and* $\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i}) := \alpha_i(\mathbf{s}, a_i)\ \sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})$.

Proof. We suppose for any $\mathbf{s} \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$, we have $\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = \sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})\ \hat{Q}_i(\mathbf{s}, a_i)$ and $\mathbb{E}_{\mathcal{C}_i}[\sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})] = 1$ when $a_i = \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i)$. By the definition of the Markov Shapley Q-value, it is not difficult to obtain

$$\begin{aligned} Q_i^\phi(\mathbf{s}, a_i) &= \mathbb{E}_{\mathcal{C}_i}[\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i)] \\ &= \mathbb{E}_{\mathcal{C}_i}[\sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})\ \hat{Q}_i(\mathbf{s}, a_i)] \\ &= \mathbb{E}_{\mathcal{C}_i}[\sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})]\ \hat{Q}_i(\mathbf{s}, a_i). \end{aligned}$$

Recall that $\delta_i(\mathbf{s}, a_i)$ is defined as follows:

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i). \end{cases}$$

If $a_i = \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i)$, it is not difficult to get that $Q_i^\phi(\mathbf{s}, a_i) = \hat{Q}_i(\mathbf{s}, a_i)$.

If $a_i \neq \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i)$, we can have the following equation such that

$$\begin{aligned} \alpha_i(\mathbf{s}, a_i)\ Q_i^\phi(\mathbf{s}, a_i) &= \alpha_i(\mathbf{s}, a_i)\ \mathbb{E}_{\mathcal{C}_i}[\sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})\ \hat{Q}_i(\mathbf{s}, a_i)] \\ &= \mathbb{E}_{\mathcal{C}_i}[\alpha_i(\mathbf{s}, a_i)\ \sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})]\ \hat{Q}_i(\mathbf{s}, a_i) \\ &:= \mathbb{E}_{\mathcal{C}_i}[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})]\ \hat{Q}_i(\mathbf{s}, a_i), \end{aligned}$$

where $\alpha_i(\mathbf{s}, a_i)\ \sigma(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})$ is defined as $\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})$. Since under this situation $\hat{Q}_i(\mathbf{s}, a_i)$ is always a scaled $Q_i^\phi(\mathbf{s}, a_i)$ with the scale of $1/K$, the decisions are consistent to the original decisions. □

E.6.1 Implementation of $\hat{\alpha}_i(\mathbf{s}, a_i)$

As introduced in the main part of paper, when $a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i)$, $\hat{\alpha}_i(\mathbf{s}, a_i)$ is implemented as follows:

$$\hat{\alpha}_i(\mathbf{s}, a_i) = \frac{1}{M}\sum_{k=1}^{M} F_{\mathbf{s}}\Big(\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}),\ \hat{Q}_i(\tau_i, a_i)\Big) + 1,$$

where

$$\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}) = \frac{1}{|\mathcal{C}_i^k|}\sum_{j\in\mathcal{C}_i^k} \hat{Q}_j(\tau_j, a_j)$$

and $\mathcal{C}_i^k \sim Pr(\mathcal{C}_i|\mathcal{N}\setminus\{i\})$ that follows the distribution w.r.t. the occurrence frequency of $\mathcal{C}_i$; and $F_{\mathbf{s}}(\cdot,\cdot)$ is a monotonic function with an absolute activation function on the output whose weights are generated from hypernetworks w.r.t. the global state, similar to the architecture of QMIX [9]. Since $F_{\mathbf{s}}(\cdot,\cdot) \geq 0$ always holds, it is not difficult to obtain that $\hat{\alpha}_i(\mathbf{s}, a_i) \geq 1$ always holds. As Eq.11 shows, it is not difficult to get that $\alpha_i(\mathbf{s}, a_i) = K^{-1}\ \hat{\alpha}_i(\mathbf{s}, a_i)$. Since $K \in (0,1)$, we get that $\alpha_i(\mathbf{s}, a_i) > 1$.

As introduced in the main part of paper, the following equation is satisfied such that

$$\delta_i(\mathbf{s}, a_i) = \frac{1}{|\mathcal{N}|\ w_i(\mathbf{s}, a_i)}.$$

For all $\mathbf{s} \in \mathcal{S}$ and $a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i)$, $\delta_i(\mathbf{s}, a_i) = \alpha_i(\mathbf{s}, a_i) > 1$. So, we can derive that

$$\begin{aligned} & w_i(\mathbf{s}, a_i) = \frac{1}{|\mathcal{N}|\, \alpha_i(\mathbf{s}, a_i)} \\ \Rightarrow & \max_{a_i} w_i(\mathbf{s}, a_i) = \max_{a_i} \frac{1}{|\mathcal{N}|\, \alpha_i(\mathbf{s}, a_i)} = \frac{1}{|\mathcal{N}|\, \min_{a_i} \alpha_i(\mathbf{s}, a_i)} < \frac{1}{|\mathcal{N}|} \\ \Rightarrow & 0 < \sum_{i \in \mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) < 1. \end{aligned}$$

For all $\mathbf{s} \in \mathcal{S}$ and $a_i = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i)$, $\delta_i(\mathbf{s}, a_i) = \hat{\delta}_i(\mathbf{s}, a_i) = 1$. So, we can derive that

$$\begin{aligned} & w_i(\mathbf{s}, a_i) = \frac{1}{|\mathcal{N}|} \\ \Rightarrow & \sum_{i \in \mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) = 1. \end{aligned}$$

Therefore, we can directly obtain that for all $\mathbf{s} \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$,

$$0 < \max_{\mathbf{s}} \Big\{ \sum_{i \in \mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \Big\} \leq 1.$$

Since $\gamma \in (0, 1)$, we can get that $\frac{1}{\gamma} > 1$. As a result, we show that for all $\mathbf{s} \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$,

$$0 < \max_{\mathbf{s}} \Big\{ \sum_{i \in \mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \Big\} < \frac{1}{\gamma}.$$

We get that our implementation of $\hat{\alpha}_i(\mathbf{s}, a_i)$ satisfies the condition in Theorem 1.

F Potential Negative Societal Impacts

Although this paper studies a fundamental theory of multi-agent reinforcement learning, if the proposed algorithm is applied to real-world applications in the future, there may still exist some potential negative societal impacts. First, since the theory does not consider robustness, it is possible that the proposed algorithm would be attacked or vulnerable to some extreme scenarios like most of machine learning models and algorithms. Fortunately, our theory is orthogonal to the robustness and it is possible to consider robustness as an extension in the future work. Another potential negative societal impacts could come from the implementation of models, e.g., policy and critic. Since these are implemented by neural networks that are known as black boxes, the reliability could be a problem. Nevertheless, this is irrelevant to the main purpose of this paper and can be improved by other related research tracks in the future.