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

Jianhong Wang [1] **Yuan Zhang** [2] **Yunjie Gu** [1 3] **Tae-Kyun Kim** [1 4]

Abstract

Value factorisation proves to be a useful technique for multi-agent reinforcement learning (MARL) in global reward game, but the underlying mechanism is not yet fully understood. This paper explores a theoretical framework for value factorisation with interpretability through Shapley value theory. We generalise Shapley value to Markov convex game called Markov Shapley value and apply it as a value factorisation method in global reward game, thanks to the equivalence between these two games. Based on the property of Markov Shapley value, we derive Shapley-Bellman optimality equation to evaluate the optimal Markov Shapley value that is corresponding to optimal joint deterministic policy. Furthermore, we propose Shapley-Bellman operator that is proved to solve Shapley-Bellman optimality equation. With stochastic approximation and some transformation, a new MARL algorithm called Shapley Q-learning (SHAQ) is yielded, the implementation of which is guided by the theoretical results of Shapley-Bellman operator and Markov Shapley value. In experiments, we show that SHAQ possesses not only superior performances on all tasks but also the interpretability that agrees with the theoretical analysis of Markov Shapley value and Shapley-Bellman operator.

1. Introduction

Cooperative game is 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 (Keviczky et al., 2007), autonomous distributed logistics (Schuldt, 2012) and distributed voltage control in power networks (Wang et al., 2021). 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 the 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 (Sukhbaatar et al., 2016; Omidshafiei et al., 2018; Kim et al., 2019); and (ii) each agent individually maximizes the distributed value, i.e. learning with (implicit) credit assignment (e.g., marginal contribution and value factorisation) (Foerster et al., 2018; Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019; Zhou et al., 2020).

By the view of non-cooperative game theory, global reward game can is equivalent to the Markov game (Shapley, 1953a) with a global reward (a.k.a. team reward). Its aim is learning 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 viewpoint, learning with value factorisation is inexplicable (Wang et al., 2020c). To clearly interpret the value factorisation, in this paper we stand by the side of cooperative game theory (Chalkiadakis et al., 2011), wherein the objective is partitioning agents into coalitions and finding a payoff distribution scheme to distribute the maximum value of each coalition. The corresponding solution is called Markov core, whereby no agents have incentives to deviate. Thus, if all agents are partitioned into one coalition (called the grand coalition), the payoff distribution scheme naturally plays the role of value factorisation for the optimal global value.

Wang et al. (2020c) extended convex game (i.e., a game model in cooperative game theory) (Chalkiadakis et al., 2011) to the dynamic scenario that is renamed as Markov convex game (MCG) in this paper for appropriateness. We construct the analytic form of Shapley value for the MCG that is rigorously proved reaching the Markov core under the grand coalition, named as Markov Shapley value. Since the optimal Markov Shapley value indicates not only the maximum global value but also that no agents have incentives to deviate from the grand coalition (i.e., the implication from the Markov core), the global reward game with value factorisation becomes explicable. Moreover, we prove that Markov Shapley value enjoys the following properties: (i) the sensitiveness to the dummy agents; (ii) the efficiency: the optimal global value is equal to the sum of optimal Markov Shapley values; and (iii) the fairness. All these 3 properties aid the interpretation of factorised values for de-

[1]Imperial College London, UK [2]University of Freiburg, Germany [3]University of Bath, UK [4]KAIST, South Korea. Correspondence to: Yunjie Gu <yg934@bath.ac.uk>.

cisions, and such interpretability is critical to the industrial applications (Wang et al., 2021).

Based on the property of efficiency, we derive Shapley-Bellman optimality equation that is an extension of Bellman optimality equation (Bellman, 1952; Sutton & Barto, 2018). Moreover, we propose Shapley-Bellman operator and prove its convergence to Shapley-Bellman optimality equation and the corresponding optimal joint deterministic policy. With stochastic approximation of Shapley-Bellman operator and some transformation, we derive an algorithm called Shapley Q-learning (SHAQ). Literally, SHAQ learns to approximate the optimal Markov Shapley value. Based on a condition, the effect of coalitions (that needs shared information) in Markov Shapley value is transferred to the correlated weights $\hat{\alpha}_i(\mathbf{s}, a_i)$ (see Eq.12) and thus Markov Shapley value becomes decentralised in order to fit the decentralised execution framework. Furthermore, the module design of $\hat{\alpha}_i(\mathbf{s}, a_i)$ in practice follows the theoretical analysis so that SHAQ is probable to converge to the optimal joint deterministic policy with some guarantees.

To evaluate SHAQ, we run experiments on two platforms that can be regarded as the global reward games such as Predator-Prey (Böhmer et al., 2020) and the multi-agent StarCraft benchmark tasks (Samvelyan et al., 2019). From the experimental results, SHAQ shows not only generally good performances on solving all tasks but also the interpretability that the state-of-the-art baselines lack. Specifically, the learned Markov Shapley Q-value can reflect the contribution of each agent (i.e., fairness), as well as identify the dummy agents and the equal distribution of optimal value, all of which verify our theoretical analysis.

2. Preliminaries

2.1. Markov Convex Game

We now define Markov convex game (MCG) following the prior work (Wang et al., 2020c) (that was called extended convex game). In MCG, the transition probability is defined as $Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, where $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$. $\mathcal{S}$ is the set of states and $\mathcal{A} = \times_{i\in\mathcal{N}}\mathcal{A}_i$ is the joint action set of the grand coalition $\mathcal{N}$ (i.e. the coalition including all agents) and $\mathcal{A}_i$ is each agent's action set. If considering any coalition $\mathcal{C} \subseteq \mathcal{N}$, the joint action set of agents belonging to $\mathcal{C}$ is denoted as $\mathcal{A}_\mathcal{C} = \times_{i\in\mathcal{C}}\mathcal{A}_i$. $V^{\pi_\mathcal{C}}(\mathbf{s}) = \mathbb{E}_{\pi_\mathcal{C}}\big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R_\tau(\mathcal{C}, \mathbf{s}, \mathbf{a}_\mathcal{C}) \mid \mathbf{S}_t = \mathbf{s}\big]$ where $\gamma \in (0, 1)$, represents the value function of a coalition $\mathcal{C}$ controlled by the policies of agents in $\mathcal{C}$ (i.e., $\pi_\mathcal{C}(\mathbf{a}_\mathcal{C}|\mathbf{s}) = \times_{i\in\mathcal{C}}\pi_i(a_i|\mathbf{s})$, shortened as $\pi_\mathcal{C}$); and $R_\tau : 2^\mathcal{N} \times \mathcal{S} \times \mathcal{A} \to [0, \infty)$ (i.e., a characteristic function) is the reward for coalition $\mathcal{C}$ at time step τ. Accordingly, $R_\tau(\mathcal{N}, \mathbf{s}, \mathbf{a})$ (for the grand coalition) is the global reward at time step τ that might be written as $R(\mathbf{s}, \mathbf{a})$ or R in the rest of paper, where the τ might be added to the subscript for identifying the time steps. The value of cumulative global rewards is denoted as $V^\pi(\mathbf{s}) \in [0, +\infty)$ and the value of empty coalition is denoted as $V^{\pi_\varnothing}(\mathbf{s}) = 0$. As for other coalitions $\mathcal{C} \subset \mathcal{N}$, $V^{\pi_\mathcal{C}}(\mathbf{s}) \in [0, +\infty)$. Similarly, the global Q-value (for the grand coalition) is defined as $Q^\pi(\mathbf{s}, \mathbf{a}) \in [0, +\infty)$, and the other coalition Q-value (for a coalition $\mathcal{C} \subset \mathcal{N}$) is defined as $Q^{\pi_\mathcal{C}}(\mathbf{s}, \mathbf{a}_\mathcal{C}) \in [0, +\infty)$. 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 $Q^{\pi_\mathcal{C}}_{\pi^*_\mathcal{D}}(\mathbf{s}, \mathbf{a}_\mathcal{C})$. Therefore, the optimal coalition Q-value of $\mathcal{C}$ w.r.t. the optimal joint policy of $\mathcal{C}$ is defined as $Q^{\pi_\mathcal{C}}_{\pi^*_\mathcal{C}}(\mathbf{s}, \mathbf{a})$ that is also denoted as $Q^{\pi^*_\mathcal{C}}(\mathbf{s}, \mathbf{a})$ for conciseness. Accordingly, the optimal global Q-value (w.r.t. the optimal joint policy of the grand coalition) is denoted as $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$. The solution of MCG is a tuple containing a collection of coalitions called coalition structure and a payoff distribution scheme that distributes the optimal coalition value to agents. Mathematically, this tuple is represented as $\big\langle \mathcal{CS}, \big(\max_{\pi_i} x_i(\mathbf{s})\big)_{i\in\mathcal{N}}\big\rangle$, where $\mathcal{CS} = \{\mathcal{C}_1, ..., \mathcal{C}_n\}$ is the coalition structure and $\big(\max_{\pi_i} x_i(\mathbf{s})\big)_{i\in\mathcal{N}}$ is a payoff distribution scheme. There is a condition for characterizing MCG[1] as follows:

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

Usually, we assume that $\mathcal{C}_m \cap \mathcal{C}_k = \varnothing, \forall \mathcal{C}_m, \mathcal{C}_k \subset \mathcal{N}$ and this simplifies Eq.1 to the definition in Wang et al. (2020c) where $\max_{\pi_{\mathcal{C}_\cap}} V^{\pi_{\mathcal{C}_\cap}}(\mathbf{s}) = 0$ (since $V^{\pi_\varnothing}(\mathbf{s}) = 0$).

2.2. Markov Core

In the MCG with the grand coalition (i.e., $\mathcal{CS} = \{\mathcal{N}\}$), the Markov core is defined as a set of payoff distribution schemes by which no agents have incentives to deviate from the grand coalition to gain more profits. Mathematically, Markov core can be expressed as follows:

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

wherein $\max_{\pi_\mathcal{C}} x(\mathbf{s}|\mathcal{C}) = \sum_{i\in\mathcal{C}} \max_{\pi_i} x_i(\mathbf{s})$. The definition of Markov core here is a direct extension from the original definition for static convex game (Shapley, 1971). We aim to find out a value factorisation method (i.e., $(x_i(\mathbf{s}))_{i\in\mathcal{N}}$) that can finally reach a payoff distribution scheme in the Markov core during learning process (that might be called reaching the Markov core for conciseness).

[1]The condition is actually supermodularity, but it is called convexity in the context of cooperative game theory.

3. Markov Shapley Value

In this section, we first introduce coalition marginal contribution by the view of cooperative game theory, based on which we then define Markov Shapley value.

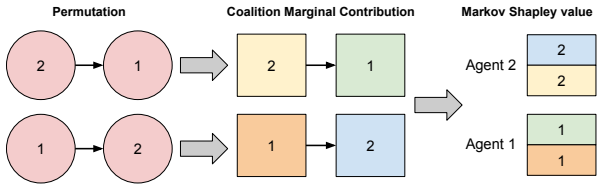


Figure 1: The illustration of calculating Markov Shapley value for the case of 2 agents. Left: Each red circle means an agent to form a permutation. Middle: Each square means the coalitional marginal contribution of an agent obtained during a permutation. Right: Each square in mixing colors means the Markov Shapley value of an agent.

By the view of cooperative game theory, the grand coalition is progressively formed by a permutation of agents. Accordingly, coalition marginal contribution is the contribution of agent i to the intermediate coalition that it would join in, as 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 coalition 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. *The coalition marginal contribution w.r.t. the action of each agent can be derived as follows:*

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

As Proposition 1 shows, The coalition marginal contribution w.r.t. the action of each agent (analogous to Q-value) can be derived according to Eq.3. It is usually more useful in solving MARL problems.

Markov Shapley Value. It is apparent that coalition marginal contribution only considers one permutation to form the grand coalition. By the viewpoint from Shapley (1953b), the fairness is achieved through considering how much the agent i increases the maximum values (i.e., the coalition marginal contributions) of all possible coalitions 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 the Shapley value based on coalition marginal contributions as Definition 2 shows, named as Markov Shapley value.

Definition 2. *Markov Shapley value is represented as*

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

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

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

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. $\frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}$ can be seen as the probability measure over $\mathcal{C}_i$, i.e., $Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$, and therefore the Markov Shapley Q-value can be interpreted as the expectation of coalition marginal contributions w.r.t. $\mathcal{C}_i \sim Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ such that

$$Q_i^{\phi}(\mathbf{s}, a_i) = \mathbb{E}_{\mathcal{C}_i \sim Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})}[\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i)]. \tag{7}$$

For better understanding the process of computing Markov Shapley value, we provide a simple illustration in Figure 1.

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

Proposition 2 shows 3 properties of Markov Shapley value. The most important feature is (ii) that aids the formulation of Shapley-Bellman optimality equation. Property (iii) provides a fundamental mechanism to quantitatively describe "fairness" among agents. Property (i) and (iii) play important roles in interpretation.

4. Shapley Q-Learning

In this section, we (i) derive Shapley-Bellman optimality equation that evaluates the optimal Markov Shapley Q-value and the corresponding optimal joint deterministic policy; (ii) propose Shapley-Bellman operator and prove its convergence to Shapley-Bellman optimality equation; (iii) derive Shapley Q-learning based on Shapley-Bellman operator; (iv) explain the rationality of the formulation of Shapley-Bellman optimality equation and provide an interpretation of Shapley Q-learning in order that the global reward game with value factorisation is explicable; and (v) implement Shapley Q-learning in decentralised execution manner and design the modules in practice as per the theoretical analysis so that the convergence to the optimal joint deterministic policy is with some theoretical guarantees.

4.1. Definition and Formulation

Shapley-Bellman Optimality Equation. Based on Bellman optimality equation (Bellman, 1952) and two conditions such that (C.1) the efficiency of the Markov Shapley Q-value proved in Proposition 2; (C.2) a suppose that $Q_i^{\phi^*}(\mathbf{s}, a_i) = w_i(\mathbf{s}, a_i)\ Q^{\pi^*}(\mathbf{s}, \mathbf{a}) - b_i(\mathbf{s})$, where $w_i > 0$ and $b_i \geq 0$ are bounded and $\sum_{i\in\mathcal{N}} w_i^{-1} b_i = 0$, we derive an equation called Shapley-Bellman optimality equation for evaluating the optimal Markov Shapley value such that

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

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 Markov Shapley Q-value. If Eq.8 holds, the optimal Markov Shapley Q-value is achieved. Moreover, it reveals an implication that $\forall \mathbf{s} \in \mathcal{S}$ and $a_i^* = \arg\max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i)$, we have the 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 making the optimal decisions. 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. (2020a) recently from another perspective of analysis. Nevertheless, the value of $w_i(\mathbf{s}, a_i)$ for $a_i \neq \arg\max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i)$ needs to be explored (e.g., through learning).

Shapley-Bellman Operator. To find an optimal solution described by Eq.8, we now propose an operator called Shapley-Bellman operator, 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 specifically defined as follows:

$$\Upsilon\big(\times_{i\in\mathcal{N}} Q_i^{\phi}(\mathbf{s}, a_i)\big) = \mathbf{w}(\mathbf{s}, \mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \big[R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i)\big] - \mathbf{b}(\mathbf{s}), \tag{9}$$

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 the Shapley-Bellman operator 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* $\max_{\mathbf{s}} \big\{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \big\} < \frac{1}{\gamma}$.

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

$$\Delta(\mathbf{s}, \mathbf{a}, \mathbf{s}') = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\ Q_i^{\phi}(\mathbf{s}, a_i), \tag{10}$$

where

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i). \end{cases} \tag{11}$$

Actually, the closed-form expression of $\delta_i(\mathbf{s}, a_i)$ is written as $\frac{1}{|\mathcal{N}|\ w_i(\mathbf{s}, a_i)}$. 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 $a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i)$ for easy implementation, Eq.11 is obtained. The term $\mathbf{b}(\mathbf{s})$ is cancelled in Eq.10 thanks to the condition such that $\sum_{i\in\mathcal{N}} w_i^{-1} b_i = 0$. Moreover, 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 for details).

4.2. Rationality and Interpretability

In this section, we verify the rationality of Shapley-Bellman optimality equation and the interpretability of SHAQ, i.e., providing the reasons why Shapley-Bellman optimality equation is rational to be formulated and SHAQ is an interpretable value factorisation method for 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.*

Shapley-Bellman Optimality Equation. First, we give a proof for showing that the optimal Markov Shapley value is a solution in the Markov core under the grand coalition. Since a solution in the Markov core implies the maximum global value (see Remark 3 in Appendix D.2.2), we can get that the optimal Markov Shapley value can lead to the maximum global value (a.k.a. social welfare). The above evidences directly motivate applying condition C.1. Condition C.2 well defines the validity of the relationship between the optimal Markov Shapley Q-value and the optimal global Q-value for the case of dummy agents (see Appendix E.4.1 for details), so that the definition of the Shapley-Bellman optimality equation is consistent with the definition of MCG and the properties of Markov Shapley Q-value.

SHAQ. The above discussion implies that solving Shapley-Bellman optimality equation is equivalent to solving the Markov core under the grand coalition. As a result, SHAQ is actually a learning algorithm that learns to approximate the optimal Markov Shapley Q-value (i.e., equivalent to that the Shapley-Bellman optimality equation holds) and

therefore reach the Markov core also holds. As per the definition in Section 2.2, 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.

4.3. Implementation in Practice

We now describe a practical implementation of SHAQ for Dec-POMDP (Oliehoek, 2012) (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 (Oliehoek, 2012). 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) (Oliehoek et al., 2008) is applied, the global state (i.e., $\mathbf{s}$) for $\hat{\alpha}_i(\mathbf{s}, a_i)$ can be obtained during training.

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 Proposition 8 (see Appendix E.6), with specific conditions the information of coalition formation can be equivalently transferred to $\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})$. As a result, $\delta_i(\mathbf{s}, a_i)$ is equivalent to the form as follows:

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

wherein $\hat{\alpha}_i(\mathbf{s}, a_i) = \mathbb{E}_{\mathcal{C}_i \sim Pr(\mathcal{C}_i | \mathcal{N}\setminus\{i\})}[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})]$ and $Pr(\mathcal{C}_i | \mathcal{N}\setminus\{i\})$ is that we introduced in Section 3. To solve partial observations (i.e., learning the belief state), $\hat{Q}_i(\tau_i, a_i)$ is represented as recurrent neural network (RNN) with GRUs (Chung et al., 2014). $\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})$ is approximated by a parametric function $F_{\mathbf{s}} + 1$ and thus $\hat{\alpha}_i(\mathbf{s}, a_i)$ can be expressed as the following equation:

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

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 $\mathcal{C}_i^k \sim Pr(\mathcal{C}_i | \mathcal{N}\setminus\{i\})$ is sampled M times (via Monte Carlo method) to approximate $\mathbb{E}_{\mathcal{C}_i}[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})]$; and $F_{\mathbf{s}}$ is a monotonic function, additionally with an absolute activation function on the output, whose weights are generated from hyper-networks w.r.t. the global state, similar to the architecture of QMIX (Rashid et al., 2018). We show that Eq.13 satisfies the condition to $w_i(\mathbf{s}, a_i)$ in Theorem 1 and therefore the practical implementation of SHAQ can converge to the optimal joint deterministic policy with some guarantees (see Appendix E.6.1 for details).

By using the framework of fitted Q-learning (Ernst et al., 2005) to solve large number of states (i.e., could be usually infinite) and inserting the above constructed modules, the practical least-square-error loss function derived from Eq.10 is therefore stated as follows:

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

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 (Mnih et al., 2013), 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 learner (Claus & Boutilier, 1998), centralised training and decentralised execution (CTDE) (Oliehoek et al., 2008) was proposed and became a general paradigm for MARL. Based on CTDE, MADDPG learns a global Q-value that can be regarded as identically assigning to all agents with the same credits during training (Wang et al., 2020c), which may cause the unfair credit assignment (Wolpert & Tumer, 2002). To avoid this problem, VDN (Sunehag et al., 2018) was proposed to learn the factorised Q-value, assuming that any global Q-value is equal to the summation of decentralised Q-values. Nevertheless, this factorisation of the global Q-value may cause the limitation on representation of the global Q-value. To mitigate this issue, QMIX (Rashid et al., 2018) and QTRAN (Son et al., 2019) were proposed to represent the global Q-value with a richer class w.r.t. decentralised Q-values, based on an assumption called Individual-Global-Max (IGM) that was for promising the convergence to the optimal joint deterministic policy. Markov Shapley value proposed in this paper belongs to the family of value factorisation, but based on the game-theoretical framework called MCG that enjoys the interpretability. With the realistic insights from the conventional cooperative games (e.g., network flow games (Kalai & Zemel, 1982), induced subgraph game (Deng & Papadimitriou, 1994) that can be used for modelling social networks, and facility location games (Deng et al., 1999)), it is evident that the coalition introduced in this paper exists with the realistic insights. In many scenarios, the information of coalition might be unknown. To address it, it can be naturally assumed that the coalition inherently exists and we only need to concentrate on the exposed information, e.g., the global reward.

Relationship to VDN. By setting $\delta_i(\mathbf{s}, a_i) = 1$ for any state-action pairs, SHAQ degrades to VDN (Sunehag et al., 2018). Although solving the problem of dummy agents is a motivation of VDN, Sunehag et al. (2018) did not give a theoretical

guarantee on identifying the dummy agents. The Markov Shapley value theory proposed in this paper well addressed this issue from both theoretical and empirical aspects. These two evidences strongly show that VDN is a subclass of SHAQ. Accordingly, the theoretical framework proposed in this paper gives the answer for why VDN works well in most scenarios and why it performs poorly on some scenarios (i.e., $\delta_i(\mathbf{s}, a_i) = 1$ in Eq.10 was incorrectly defined over the suboptimal actions).

Comparison with SQDDPG. Since this work is an extension from Wang et al. (2020c), we clarify the difference between these two works to avoid the confusions. (i) This paper proposes Markov Shapley value that consolidates and corrects the analytic form of Shapley Q-value (Wang et al., 2020c).[2] (ii) The Shapley-Bellman optimality equation and Shapley-Bellman operator proposed in this paper forms the theoretical analysis of finding the optimal Markov Shapley Q-value, which did not appear in Wang et al. (2020c). Owing to the above novel framework extension, SHAQ possesses a different loss function from SQDDPG (Wang et al., 2020c) to update Markov Shapley Q-value. (iii) SHAQ is implemented in a decentralised manner to approximate the optimal Markov Shapley Q-value to fit the paradigm of multi-agent Q-learning, while SQDDPG is implemented in a centralised manner and difficult to be applied to multi-agent Q-learning directly.

Shapley Value for Machine Learning. Shapley value has been broadly applied in machine learning research community. Lundberg & Lee (2017), Ancona et al. (2019) and Kumar et al. (2020) applied Shapley value as a measure of feature importance for statistical models or deep neural networks. Jia et al. (2019) valued annotated data by approximating their contributions to the model. These works above lie in static scenarios that just directly leveraged the conventional Shapley value theory for application. However, this paper studies the extension of the Shapley value theory to Markov dynamics.

6. Experiments

In this section, we show the experimental results of SHAQ on Predator-Prey (Böhmer et al., 2020) and various tasks in StarCraft Multi-Agent Challenge (SMAC) [3]. The baselines that we select for comparison are COMA (Foerster et al., 2018), VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2018), MASAC (Iqbal & Sha, 2019), QTRAN (Son et al., 2019), QPLEX (Wang et al., 2020b) and W-QMIX (including CW-QMIX and OW-QMIX) (Rashid et al., 2020). The implementation details of our algorithm are shown in Appendix B.1, whereas the implementation of baselines are from (Rashid et al., 2020) [4]. We also compare SHAQ with SQDDPG (Wang et al., 2020c) [5], which is left to Appendix C.4 due to the limitation of space. 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 (Wang et al., 2020b). For the easy and hard maps in SMAC, we apply 50k time steps for annealing, the same as that in (Samvelyan et al., 2019); 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 (Rashid et al., 2020). To fairly evaluate all algorithms, we run each experiment with 5 random seeds. All graphs showing experimental results are plotted with the median and 25%-75% quartile shading.

6.1. Predator-Prey

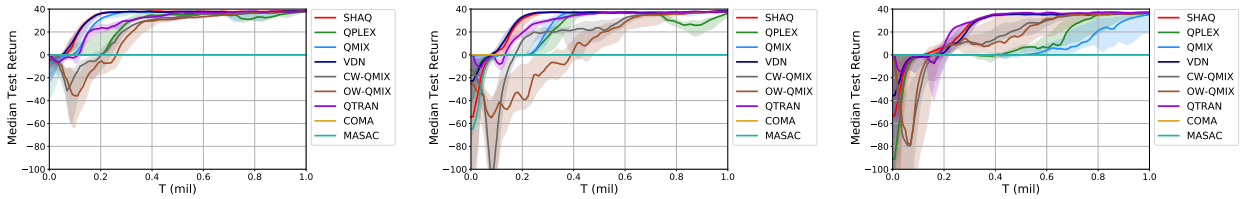
In this section, we run the experiments on a partially-observable task called Predator-Prey (Böhmer et al., 2020), wherein 8 predators that we can control aim to capture 8 preys with random policies in a 10x10 grid world. Each agent's observation is a 5x5 sub-grid centering around it. If a prey is captured 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 (Rashid et al., 2020) reported, only QTRAN and W-QMIX can solve this task, while (Wang et al., 2020b) 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) employed in (Wang et al., 2020b). As Figure 2 shows, SHAQ can solve the tasks with different values of p. With the epsilon annealing strategy from (Wang et al., 2020b), W-QMIX does not perform as well as reported in (Rashid et al., 2020). The reason could be its poor robustness to the increased explorations (Rashid et al., 2020) 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 rel-

[2] Markov Shapley Q-value defined in this paper is the corrected form of Shapley Q-value.

[3] The version that we use in this paper is SC2.4.6.2.69232 rather than the newer SC2.4.10. As reported from (Rashid et al., 2020), the performance is not comparable across versions.

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

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



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

Figure 2: Median test return for Predator-Prey with different values of p.

ative overgeneralisation[6] prevents policy gradient methods from better coordination (Wei et al., 2018).

6.2. StarCraft Multi-Agent Challenge

We now evaluate SHAQ on the more challenging SMAC tasks, the environmental settings of which are the same as that in (Samvelyan et al., 2019). 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.2.

Performance Analysis. First, we compare performances between SHAQ and baselines. 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 (Son et al., 2019), 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 the lack of the law on hyperparameter tuning (Rashid et al., 2020) it is difficult to be adapted for all scenarios (see Appendix C.3 for details).

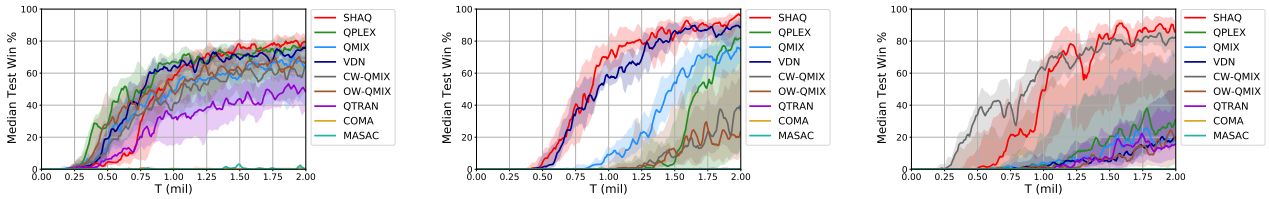
Interpretability of SHAQ. To show the interpretability of SHAQ, we conduct a test on 3m (i.e., a simple task in SMAC), demonstrating the learned Markov Shapley Q-values of both ϵ-greedy policy (for obtaining mixed optimal and suboptimal decisions) and greedy policy (for obtaining optimal decisions). 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. The Markov Shapley Q-value 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 Markov Shapley Q-value is greater, which reflects the fairness (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 Markov Shapley Q-values (verifying the theoretical results in Section 4.1). The above results well verify the theoretical analysis that we deliver.

To verify that the Markov Shapley Q-values learned by SHAQ is non-trivial, we also show the results of VDN, QMIX and QPLEX. It is surprising that the decentralised Q-values of all baselines are also almost identical among agents for the optimal decisions (however, the property of baselines disappears in more complicated scenarios as shown in Appendix C.5). 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. The explanation for the results of QMIX and QPLEX deserves to be studied in the future work. As for the suboptimal decisions, 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). Similarly, QMIX and QPLEX cannot show explicit interpretations over suboptimal decisions either. To enable the conclusion to be more convincing, we also visualise the results on the complicated 3s5z_vs_3s6z (see Appendix C.5 for details).

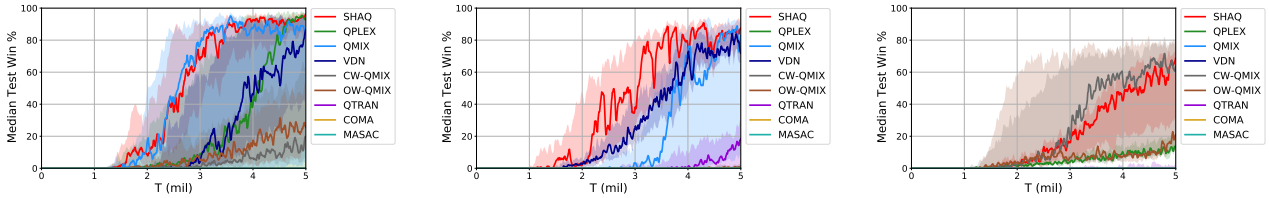
6.3. Ablation Study

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

[6]Relative overgeneralisation is a common game theoretic pathology that the suboptimal actions are preferred when matched with arbitrary actions from the collaborating agents (Wei & Luke, 2016).



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



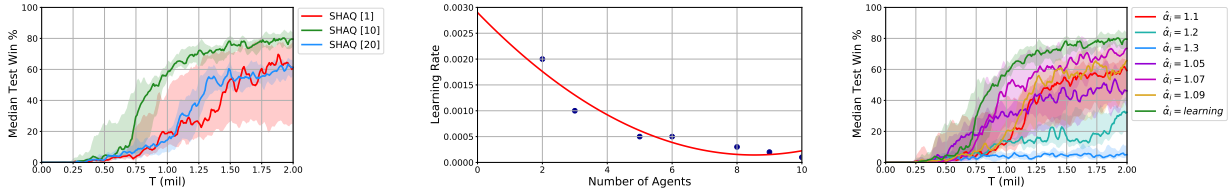
(d) 3s5z_vs_3s6z. (e) Corridor. (f) 6h_vs_8z.

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

(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.



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

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

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

Figure 5: The figures of 3 ablation studies of SHAQ on SMAC.

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 5a 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 (Byrd et al., 2012; Hofmann et al., 2015). As a

result, we select M = 10 in practice, to trade off between convergence rate and variance.

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 5b. 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 5b.

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 5c 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.

7. Conclusion

This paper generalises Shapley value to Markov convex game (MCG), called Markov Shapley value. Markov Shapley value possesses 3 properties: (i) the sensitiveness to dummy agents; (ii) the efficiency; and (iii) the fairness. Based on property (ii), we derive Shapley-Bellman optimality equation, Shapley-Bellman operator and Shapley Q-learning (SHAQ). We prove that solving Shapley-Bellman optimality equation is equivalent to solving the Markov core (i.e., no agents have incentives to deviate from the grand coalition). MCG with the grand coalition is equivalent to the global reward game (Wang et al., 2020c), wherein Markov Shapley value plays the role of value factorisation. Since SHAQ is a sort of stochastic approximation of Shapley-Bellman operator that is shown to solve Shapley-Bellman optimality equation, the global reward game with value factorisation becomes explicable standing by the proposed theoretical framework in this paper. Property (i) and (iii) aid the interpretation of decisions. We evaluate the performance of SHAQ on Predator-Prey (Böhmer et al., 2020) and the challenging multi-agent StarCraft benchmark tasks (Samvelyan et al., 2019). Several theoretical claims (especially property (i) and (iii)) are verified, while SHAQ shows the generally good performances compared with the state-of-the-art baselines. In the future work, we will extend the work when the MCG does not hold and derive an algorithm that can still achieve the Markov core. Another thread is investigating the invariant representation of $\hat{\psi}_i$ such as using graph neural networks, so that both interpretability and performance can be improved further.

Acknowledgements

This work is supported by the Engineering and Physical Sciences Research Council of UK (EPSRC) under awards EP/S000909/1.

References

Ancona, M., Oztireli, C., and Gross, M. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pp. 272–281. PMLR, 2019.

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

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

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

Byrd, R. H., Chin, G. M., Nocedal, J., and Wu, Y. Sample size selection in optimization methods for machine learning. *Mathematical Programming*, 134(1):127–155, 2012. doi: 10.1007/s10107-012-0572-5.

Chalkiadakis, G., Elkind, E., and Wooldridge, M. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

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

Dales, H. G., Dales, H. G., Aiena, P., Eschmeier, J., Laursen, K., and Willis, G. A. *Introduction to Banach algebras, operators, and harmonic analysis*, volume 57. Cambridge University Press, 2003.

Deng, X. and Papadimitriou, C. H. On the complexity of cooperative solution concepts. *Mathematics of operations research*, 19(2):257–266, 1994.

Deng, X., Ibaraki, T., and Nagamochi, H. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Ha, D., Dai, A. M., and Le, Q. V. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Hofmann, T., Lucchi, A., Lacoste-Julien, S., and McWilliams, B. Variance reduced stochastic gradient descent with neighbors. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28, pp. 2305–2313. Curran Associates, Inc., 2015.

Iqbal, S. and Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 2961–2970. PMLR, 2019.

Jaakkola, T., Jordan, M. I., and Singh, S. P. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176. PMLR, 2019.

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

Keviczky, T., Borrelli, F., Fregene, K., Godbole, D., and Balas, G. J. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on control systems technology*, 16(1):19–33, 2007.

Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., and Yi, Y. Learning to schedule communication in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pp. 5491–5500. PMLR, 2020.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Melo, F. S. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Oliehoek, F. A. Decentralized pomdps. In *Reinforcement Learning*, pp. 471–503. Springer, 2012.

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

Omidshafiei, S., Kim, D.-K., Liu, M., Tesauro, G., Riemer, M., Amato, C., Campbell, M., and How, J. P. Learning to teach in cooperative multiagent reinforcement learning. *arXiv preprint arXiv:1805.07830*, 2018.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. 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*, pp. 4292–4301. PMLR, 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

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

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

Shapley, L. S. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953a.

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

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

Son, K., Kim, D., Kang, W. J., Hostallero, D., and Yi, Y. 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*, pp. 5887–5896. PMLR, 2019.

Sukhbaatar, S., szlam, a., and Fergus, R. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems 29*, pp. 2244–2252. Curran Associates, Inc., 2016.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., and Graepel, T. 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*, pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Wang, J., Ren, Z., Han, B., Ye, J., and Zhang, C. Towards understanding linear value decomposition in cooperative multi-agent q-learning. *arXiv preprint arXiv:2006.00587*, 2020a.

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

Wang, J., Zhang, Y., Kim, T.-K., and Gu, Y. 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 2020c.

Wang, J., Xu, W., Gu, Y., Song, W., and Green, T. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in Neural Information Processing Systems*, 34, 2021.

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

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

Wolpert, D. H. and Tumer, K. Optimal payoff functions for members of collectives. In *Modeling complexity in economic and social systems*, pp. 355–369. World Scientific, 2002.

Zhou, M., Liu, Z., Sui, P., Li, Y., and Chung, Y. Y. Learning implicit credit assignment for multi-agent actor-critic. *arXiv preprint arXiv:2007.02529*, 2020.

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

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

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

    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}
    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 $\mathcal{E}$
  end for
  Store $\mathcal{E}$ to $\mathcal{B}$
  Sample a batch of episodes with batch size B from $\mathcal{B}$
  for each sampled episode **do**
    for k=1:T **do**
      Get each transition $\left\langle \mathbf{s}^k, (o_i^k)_{i=1:N}, (a_i^k)_{i=1:N}, R^k, \mathbf{s}^{k+1}, (o_i^{k+1})_{i=1:N} \right\rangle$
      For each agent i, get $\tau_i^k = (o_i^m)_{m=1:k}$
      For each agent i, calculate $\hat{Q}_i(\tau_i^k, a_i^k; \theta)$
      For each agent i, calculate $\alpha_i(\mathbf{s}^k, a_i^k; \lambda)$ by Algorithm 2
      For each agent i, calculate $\delta_i(\mathbf{s}^k, a_i^k; \lambda)$ as follows:

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

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

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

  Update θ and λ through the above loss
  Periodically update θ^- by copying θ
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 before, 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 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, M coalitions for agent i are obtained for calculating the coalition marginal contributions and therefore the approximate Markov Shapley value is obtained.

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 (Ha et al., 2017) 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 (Chung et al., 2014), 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 (Goodfellow et al., 2014), 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 with periods from 4 to 26 hours.

B.2. Hyperparameters of Baselines

The hyperparameters of all baselines except for SQDDPG (Wang et al., 2020c) are consistent with Rashid et al. (2020) and Wang et al. (2020b). 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 (Böhmer et al., 2020) in Table 2.

B.4. StarCraft Multi-Agent Challenge

The StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) 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, 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

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

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

C. Extra Experimental Results

C.1. Ablation Study of SHAQ

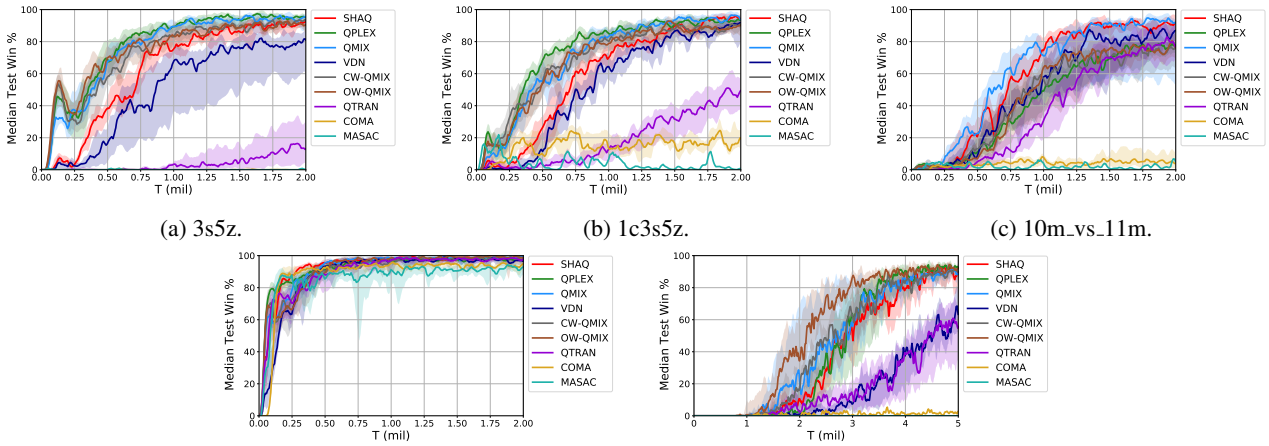
C.2. Experimental Results on Extra SMAC Maps

To thoroughly compare the performance of SHAQ with baselines, we also run experiments on 5 extra maps in SMAC as Figure 6 shows. 8m, 3s5z, 1c3s5z and 10m_vs_11m are an easy maps and MMM2 is a super-hard map. The strategy of

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

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.

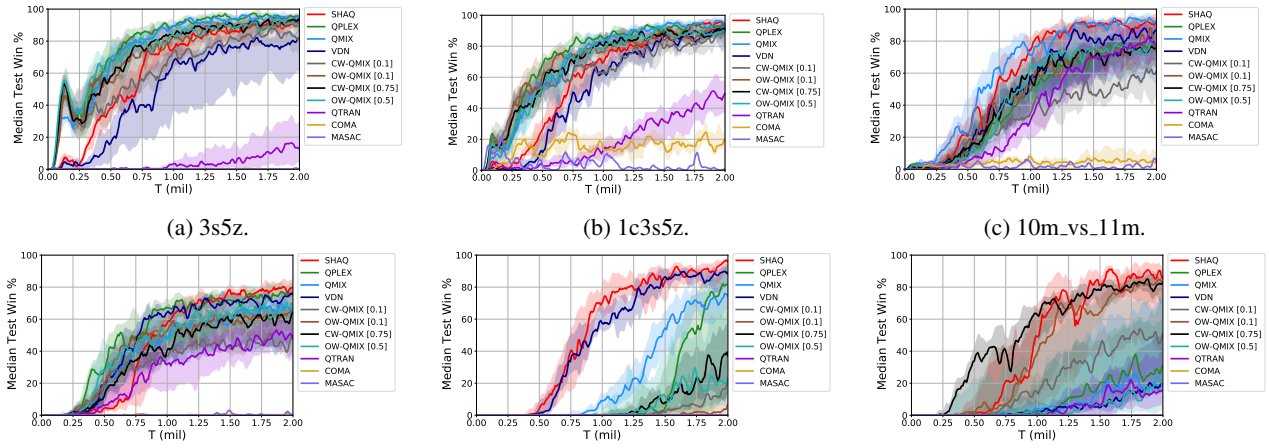
Figure 6: Median test win % for 5 extra maps in SMAC.

C.3. 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 (Rashid et al., 2020). We can observe from Figure 7 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. (2020) 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 5c shows.

C.4. Comparison with SQDDPG

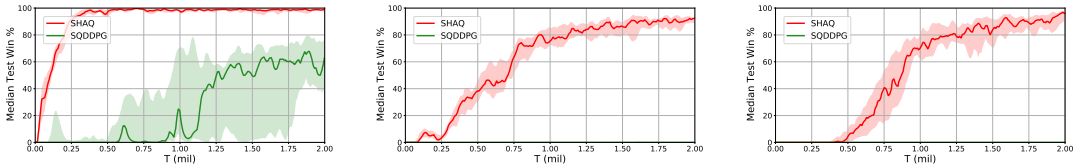
To emphasize the improvement of SHAQ from SQDDPG (Wang et al., 2020c), we exclusively compare these two algorithms on 3 maps in SMAC. As Figure 8 shows, the performance of SHAQ surpasses that of SQDDPG on all 3 maps, while SQDDPG can only learn on the simplest map 3m. The most possible reason for the 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 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 10 shows. It is apparent that SHAQ can still outperform SQDDPG.



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

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

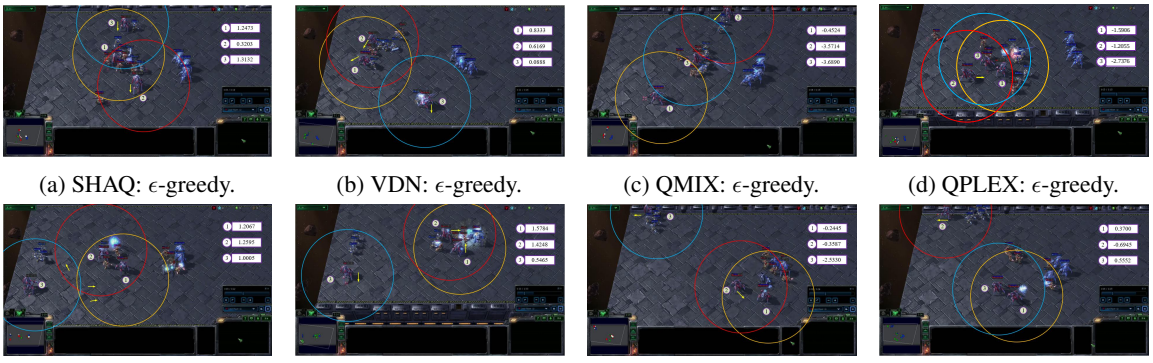
Figure 7: Median test win % for easy (1st row) and hard (2nd row) maps of SMAC for W-QMIX with different α.



(a) 3m. (b) 3s5z. (c) 3s_vs_5z.

Figure 8: Median test win % for 3 maps of SMAC to compare SHAQ with SQDDPG.

C.5. More Visualisation



(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.

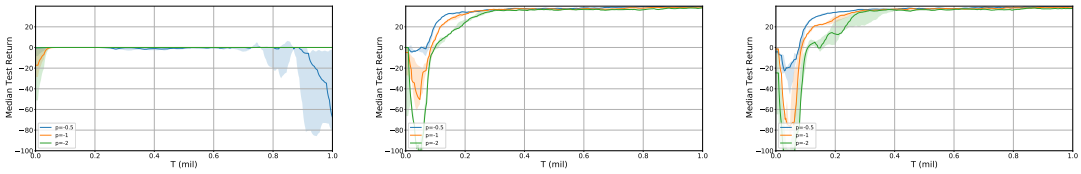
Figure 9: Visualisation of the evaluation for SHAQ and baselines on 3s5z_vs_3s6z in SMAC: each colored circle is the centered attacking range of a controllable agent (in red), and each agent's factorised Q-value is reported on the right. We mark the direction that each moving agent face by an arrow.

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 decisions. 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 decisions and suboptimal decisions. Therefore, the parametric effects of the mistakes conducted on suboptimal decisions to the optimal decisions by VDN during learning may be exaggerated when the number of agents increases. About QMIX and QPLEX, the Q-values of optimal decisions

are difficult to be interpreted in this complicated scenario. For both algorithms, the agent who is responsible for kiting [7] (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 decisions. 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 is again consistent 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 is 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 10, 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 owing 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) (Son et al., 2019) that is defined in Definition 3.

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*

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

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

There are 3 popular frameworks that are followed by most of works implementing the IGM, called VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2018) and QTRAN (Son et al., 2019).

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

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

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 (Son

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

et al., 2019) 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 (Rashid et al., 2018).

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. (2019) 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. (2019) 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 for MCG

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 1) 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 (Shapley, 1971), also known as the snowball effect.

Remark 1. *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 the 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 in comparison under coalition $\mathcal{C}$. By Remark 2 and 3, it is obvious that Eq.2 indicates that the maximum 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 (Wang et al., 2020c). It is an intuitive interpretation of the Markov core (under the grand coalition).

Remark 2. *Suppose that a coalition structure is written as $\mathcal{CS} = \{\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_n\}$, where $\cup_{k=1}^n \mathcal{C}_k = \mathcal{N}$ and each $\mathcal{C}_k$ is mutually exclusive (i.e., $\mathcal{C}_m\cap\mathcal{C}_n=\varnothing$, 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 3. *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 $\cup_{k=1}^n\mathcal{C}_k=\mathcal{N}$ 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 2, 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 showed that the solution in the Markov core under the grand coalition is equivalent to the maximum global value.

E. Complete Mathematical Proofs

E.1. Assumptions

Assumption 1. *Any global value function is able to be approximated by a linear decomposition.*

Assumption 2. *Any joint policy can be factorised to a permutation of decentralised (i.e., disjoint) policies based on predecessor coalitions, i.e.,* $\pi_{\mathcal{C}} = \times_{i \in \mathcal{C}} \pi_i(\mathcal{C}_i)$, *where* $\mathcal{C}_i \in \Pi(\mathcal{C})$. $\Pi(\mathcal{C})$ *here is a set of predecessor coalitions induced by an arbitrary permutation (or an arbitrary permutation) of agents to form a coalition* $\mathcal{C}$. $\pi_i(\mathcal{C}_i)$ *is a policy of agent* i *w.r.t. the predecessor coalition* $\mathcal{C}_i$ *that is irrelevant to the permutation of the agents in* $\mathcal{C}_i$, *which is consistent with the definition of* $V^{\pi_{\mathcal{C}}}(\mathbf{s})$ *as a characteristic function (i.e., a set-valued function). Accordingly, an optimal global value (w.r.t. the joint policy of the grand coalition) is able to be decomposed into disjoint values in multiple forms where each form is corresponding to a sort of permutation.*

Assumption 3. *The functional space of each agent* i*'s policy* π_i *is able to be separable with respect to the predecessor coalitions, i.e.,* $\pi_i = \biguplus_{\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}} \pi_i(\mathcal{C}_i)$, $\forall i \in \mathcal{N}$, *where* $\biguplus$ *denotes the disjoint union. For example,* $\pi_i(\mathcal{C}_i) \cap \pi_i(\mathcal{D}_i) = \varnothing$, *if* $\mathcal{C}_i \neq \mathcal{D}_i$, $\forall \mathcal{C}_i, \mathcal{D}_i \in \mathcal{N} \setminus \{i\}$. *Literally, each agent is able to learn a generalised policy that is capable of handling "sub-policies" for different predecessor coalitions, where each "sub-policy" for a predecessor coalition is with a disjoint parametric space. Note: To agent* i *within only one permutation to form the grand coalition whose predecessor coalition is* $\mathcal{C}_i$, π_i *just indicates* $\pi_i(\mathcal{C}_i)$, *e.g., the notation in the proof of Lemma 7.*

Assumption 4. *If an agent* $i \in \mathcal{N}$ *is dummy, it will not provide any contribution to any predecessor coalition* $\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}$. *Additionally, no members in the predecessor coalition* $\mathcal{C}_i$ *will react in different manners after agent* i *joins.*

E.2. Preliminary Theoretical Results

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

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

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

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 2.1. □

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

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

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

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 2 and 3 we can get the equations such that

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

□

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

Proof. The complete proof is as follows.

Firstly, if we would like to prove that the optimal coalition 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 $\max_{\pi_\mathcal{C}} \Phi(\mathbf{s}|\mathcal{C}) = \sum_{i\in\mathcal{C}} \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i)$.

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 $\pi_i(\mathcal{C}_i)$ to Eq.33, we can get that

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

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

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

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 coalition marginal contribution is a solution in the Markov core of Markov convex game (MCG) with the grand coalition. □

E.3. Mathematical Proofs for the Markov Shapley Value

Proposition 1. *The coalition marginal contribution w.r.t. the action of each agent can be derived as follows:*

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

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:

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

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

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

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

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

By Eq.39, we can get the following result such that

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

The proof is completed. □

Lemma 2. *For any agent $i \in \mathcal{N}$, $\forall \mathbf{s} \in \mathcal{S}$, its optimal Markov Shapley value denoted as $\max_{\pi_i} V^{\phi}_i(\mathbf{s})$ satisfies the following equation such that*

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

where $\pi_i(\mathcal{C}_i)$ is the policy of agent i with respect to its predecessor coalition $\mathcal{C}_i$.

Proof. By convexity of the maximum operator, we can easily derive the equation such that

$$\max_{\pi_i} V^{\phi}_i(\mathbf{s}) \leq \sum_{\mathcal{C}_i\subseteq\mathcal{N}\setminus\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i). \tag{41}$$

However, if we reasonably assume that the functional space of each agent's policy is separable with respect to its predecessor coalition $\mathcal{C}_i \subseteq \mathcal{N}$ as Assumption 3 claims, we can write Eq.41 as follows:

$$\max_{\pi_i} V^{\phi}_i(\mathbf{s}) = \sum_{\mathcal{C}_i\subseteq\mathcal{N}\setminus\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i), \tag{42}$$

where $\pi_i(\mathcal{C}_i)$ is a sub-policy of agent i with respect to its predecessor coalition $\mathcal{C}_i$. □

Proposition 2. *In Markov convex game with the grand coalition, the Markov Shapley value possesses properties as follows: (i) the sensitiveness to dummy agents: $V^{\phi}_i(\mathbf{s}) = 0$; (ii) the efficiency: $\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i\in\mathcal{N}} \max_{\pi_i} V^{\phi}_i(\mathbf{s})$; (iii) the fairness.*

Proof. The complete proof is as follows. Since (iii) is actually the definition that does not need to be proved, we will firstly prove the (i), then (ii). For any agent $i \in \mathcal{N}$, $\forall \mathbf{s} \in \mathcal{S}$, its Markov Shapley value denoted as $V^{\phi}_i(\mathbf{s})$.

Proof of (i). Let us define $\Pi(\mathcal{N})$ as the set of all permutations of agents. For any permutation $m \in \Pi(\mathcal{N})$ of agents to form the grand coalition, by the reasonable assumption in Assumption 4, for any predecessor coalition $\mathcal{C}^m_i \subseteq \mathcal{N}\setminus\{i\}$ we have $\max_{\pi_{\mathcal{C}^m_i}} V^{\pi_{\mathcal{C}^m_i}}(\mathbf{s}) = \max_{\pi_{\mathcal{C}^m_i}} V^{\pi_{\mathcal{C}^m_i\cup\{i\}}}(\mathbf{s})$, $\forall \mathbf{s} \in \mathcal{S}$, thereby $\Phi_i(\mathbf{s}|\mathcal{C}^m_i) = 0$. Also, the above analysis fulfills for all permutations of agents to form the grand coalition. By Definition 2, it is not difficult to see that the Markov Shapley value for the dummy agent will be 0 such that $V^{\phi}_i(\mathbf{s}) = 0$. The proof of (i) completes.

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

$$\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i\in\mathcal{N}} \max_{\pi_i} V^{\phi}_i(\mathbf{s}), \ \forall \mathbf{s} \in \mathcal{S}, \tag{43}$$

where $V_i^{\phi}(\mathbf{s})$ denotes the Markov Shapley value. By the result from Proposition 5 and Assumption 2, for an arbitrary permutation $m \in \Pi(\mathcal{N})$ we can get the equation such that

$$\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m), \ \forall \mathbf{s} \in \mathcal{S}, \tag{44}$$

where $\Phi_i(\mathbf{s}|\mathcal{C}_i^m)$ is a coalition marginal contribution and $\pi_i(\mathcal{C}_i^m)$ ($\mathcal{C}_i^m$ is the predecessor coalition that agent i meets in the permutation m) is a sub-policy of agent $i \in \mathcal{N}$ for the permutation m. 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

$$\sum_{m \in \Pi(\mathcal{N})} \max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{m \in \Pi(\mathcal{N})} \sum_{i \in \mathcal{N}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m), \ \forall \mathbf{s} \in \mathcal{S}. \tag{45}$$

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

$$\frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi} V^{\pi}(\mathbf{s}) = \frac{1}{|\mathcal{N}|!} \sum_{i \in \mathcal{N}} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m), \ \forall \mathbf{s} \in \mathcal{S}. \tag{46}$$

Next, to ease life we start from the left hand side of Eq.46. By Assumption 2, we can directly get the following equation such that

$$\frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi} V^{\pi}(\mathbf{s}) = \frac{1}{|\mathcal{N}|!} \cdot |\mathcal{N}|! \cdot \max_{\pi} V^{\pi}(\mathbf{s}) = \max_{\pi} V^{\pi}(\mathbf{s}). \tag{47}$$

Now, we start processing the right hand side of Eq.46. By rearranging it, we can get the equations such that

$$\begin{aligned}
\frac{1}{|\mathcal{N}|!} \sum_{i \in \mathcal{N}} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) &= \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) \\
&\quad \text{(The identical } \mathcal{C}_i^m \text{ in different permutations is written as } \mathcal{C}_i \\
&\quad \text{and we can rearrange the equation as follows.)} \\
&= \sum_{i \in \mathcal{C}} \frac{1}{|\mathcal{N}|!} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}} |\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)! \cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i) \\
&= \sum_{i \in \mathcal{N}} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i).
\end{aligned} \tag{48}$$

By Lemma 2 and Assumption 3, we can get the following equations such that

$$\sum_{i \in \mathcal{N}} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i) \tag{49}$$

$$\begin{aligned}
&= \sum_{i \in \mathcal{N}} \max_{\pi_i} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \setminus \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \Phi_i(\mathbf{s}|\mathcal{C}_i) \\
&= \sum_{i \in \mathcal{N}} \max_{\pi_i} V_i^{\phi}(\mathbf{s}).
\end{aligned} \tag{50}$$

Inserting the results from Eq.47 and 50 to Eq.46, we can get the equation such that

$$\max_{\pi} V^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} V_i^{\phi}(\mathbf{s}), \ \forall \mathbf{s} \in \mathcal{S}. \tag{51}$$

Therefore, the proof for (ii) 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 (Bellman, 1952; Sutton & Barto, 2018) we can write out Bellman optimality equation for the optimal global Q-value such that

$$Q^{\pi^*}(\mathbf{s}, \mathbf{a}) = \sum_{\mathbf{s}'} Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \big[R + \gamma \max_{\mathbf{a}} Q^{\pi^*}(\mathbf{s}', \mathbf{a}) \big]. \tag{52}$$

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

$$\max_{\mathbf{a}} Q^{\pi^*}(\mathbf{s}', \mathbf{a}) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}', a_i). \tag{53}$$

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

$$Q_i^{\phi^*}(\mathbf{s}, a_i) = w_i(\mathbf{s}, a_i)\, Q^{\pi^*}(\mathbf{s}, \mathbf{a}) - b_i(\mathbf{s}). \tag{54}$$

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.54 we can write that

$$\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = \mathbf{w}(\mathbf{s}, \mathbf{a})\, Q^{\pi^*}(\mathbf{s}, \mathbf{a}) - \mathbf{b}(\mathbf{s}). \tag{55}$$

Besides, suppose that $\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}) = 0$. For an arbitrary state $\mathbf{s} \in \mathcal{S}$, it is not difficult to check that even if any agent is dummy (i.e., $Q_i^{\phi^*}(\mathbf{s}, a_i) = 0$ for some $i \in \mathcal{N}$), the optimal global Q-value $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ and $Q_j^{\phi^*}(\mathbf{s}, a_j), \forall j \neq i$ would not be forced to be zero only 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, then we only need to set $b_i(\mathbf{s}) = 0, \forall i \in \mathcal{N}$ so that the optimal global Q-value $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ would be zero and the definition of 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.

Combined with Eq.53 and 55, we can rewrite Eq.52 to the equation as follows:

$$\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = \mathbf{w}(\mathbf{s}, \mathbf{a}) \sum_{\mathbf{s}'} Pr(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \Big[R \; + \; \gamma \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}', a_i) \Big] - \mathbf{b}(\mathbf{s}). \tag{56}$$

From Eq.54, we know that $w_i(\mathbf{s}, a_i) > 0$. Therefore, we can rewrite Eq.54 to the following equation such that

$$w_i(\mathbf{s}, a_i)^{-1} \left(Q_i^{\phi^*}(\mathbf{s}, a_i) + b_i(\mathbf{s}) \right) = Q^{\pi^*}(\mathbf{s}, \mathbf{a}). \tag{57}$$

If we sum up Eq.57 for all agents, we can obtain that

$$\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} \left(Q_i^{\phi^*}(\mathbf{s}, a_i) + b_i(\mathbf{s}) \right) = |\mathcal{N}|\, Q^{\pi^*}(\mathbf{s}, \mathbf{a}). \tag{58}$$

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

$$\sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi^*}(\mathbf{s}, a_i) = \; Q^{\pi^*}(\mathbf{s}, \mathbf{a}). \tag{59}$$

Substituting Eq.59 for $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ in Eq.53, we can get the following equation such that

$$\max_{\mathbf{a}} \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|\, w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi^*}(\mathbf{s}, a_i) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i). \tag{60}$$

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

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

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 the solution $w_i(\mathbf{s}, a_i^*) = 1/|\mathcal{N}|$.

E.4.2. PROOF OF THEOREM 1

Lemma 3 (Dales et al. (2003))**.** *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 4. *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:

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

Therefore, we prove that 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. □

Lemma 5. *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 4, $||\cdot||_1$ defined here is a sub-multiplicative norm. By Lemma 3, 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 $\mathbf{Q}_1^\phi = \times_{i\in\mathcal{N}}\big(Q_i^\phi\big)_1 \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{S}||\mathcal{A}|}$ and $\mathbf{Q}_2^\phi = \times_{i\in\mathcal{N}}\big(Q_i^\phi\big)_2 \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{S}||\mathcal{A}|}$, 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}) \\
&- \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 \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.)} \\
&= \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 (Banach, 1922), Shapley-Bellman operator admits a unique fixed point. Moreover, starting by an arbitrary start point, the permutation 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 5, by Banach fixed-point theorem (Banach, 1922) we can directly conclude that Shapley-Bellman operator Υ admits a unique fixed point. Furthermore, starting by an arbitrary start point, the permutation 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.8) 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 is corresponding 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 property of efficiency 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:

$$\mathbf{Q}^{\phi}_{t+1}(\mathbf{s}, \mathbf{a}) \leftarrow \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) + \alpha_t(\mathbf{s}, \mathbf{a}) \Big[\mathbf{w}(\mathbf{s}, \mathbf{a}) \Big(R_t + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} (Q^{\phi}_i)_t(\mathbf{s}', a_i) \Big) - \mathbf{b}(\mathbf{s}) - \mathbf{Q}^{\phi}_{t}(\mathbf{s}, \mathbf{a}) \Big]. \tag{62}$$

Lemma 6 (Jaakkola et al. (1994))**.** *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 MCG, the Q-learning algorithm derived by Shapley-Bellman operator given by the update rule such that*

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

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

$$\sum_t \alpha_t(\mathbf{s}, \mathbf{a}) = \infty \qquad \sum_t \alpha_t^2(\mathbf{s}, \mathbf{a}) \le \infty \tag{63}$$

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

Proof. The proof follows the sketch of proving the convergence of Q-learning given by Melo (2001). First, we rewrite Eq.62 to

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

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.62 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

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

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

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

By the results from Theorem 5, 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

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

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 6 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.63. □

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:

$$\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = \mathbf{w}(\mathbf{s},\mathbf{a})\big(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(s,a_i)\big) - \mathbf{b}(\mathbf{s}), \tag{64}$$

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 $\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = [Q_i^{\phi}(s,a_i)]^{\top} \in \mathbb{R}_+^{|\mathcal{N}|}$. 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.64 can be equivalently represented as

$$\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = diag\big(\mathbf{w}(\mathbf{s},\mathbf{a})\big)\,\mathbf{1}\,\big(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(s,a_i)\big) - \mathbf{b}(\mathbf{s}). \tag{65}$$

[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.65 such that

$$diag\big(\mathbf{w}(\mathbf{s},\mathbf{a})\big)^{-1}\mathbf{Q}^{\phi}(\mathbf{s},\mathbf{a}) = \mathbf{1}\left(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i)\right) - diag\big(\mathbf{w}(\mathbf{s},\mathbf{a})\big)^{-1}\mathbf{b}(\mathbf{s}). \tag{66}$$

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

$$\sum_{i\in\mathcal{N}} \frac{1}{w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi}(s, a_i) = |\mathcal{N}|\left(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i)\right) - \sum_{i\in\mathcal{N}} w_i(\mathbf{s}, a_i)^{-1} b_i(\mathbf{s}). \tag{67}$$

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 can get that

$$\sum_{i\in\mathcal{N}} \frac{1}{|\mathcal{N}| w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi}(s, a_i) = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i). \tag{68}$$

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

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \end{cases} \tag{69}$$

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 easy implementation.

By substituting Eq.69 to Eq.68, we can get the following equation such that

$$\sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\, Q_i^{\phi}(\mathbf{s}, a_i) = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i). \tag{70}$$

Therefore, we derive the TD error for Shapley Q-learning (SHAQ) such that

$$\Delta(\mathbf{s}, \mathbf{a}, \mathbf{s}') = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\, Q_i^{\phi}(\mathbf{s}, a_i). \tag{71}$$

The TD error for SHAQ is necessary for the TD error for Eq.62 (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 for Rationality and Interpretability

Lemma 7. *The 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 coalition marginal contributions. We know that Markov core is a convex set by Lemma 7 and the optimal coalition 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 8. *Suppose any coalition marginal contribution can be factorised to the form such that* $\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = m(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})\ \hat{Q}_i(\mathbf{s}, a_i)$*, with the condition such that*

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

we have

$$\begin{cases} Q_i^\phi(\mathbf{s}, a_i) = \hat{Q}_i(\mathbf{s}, a_i) & a_i = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i)\ Q_i^\phi(\mathbf{s}, a_i) = \hat{\alpha}_i(\mathbf{s}, a_i)\ \hat{Q}_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \end{cases} \tag{72}$$

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

Proof. We suppose for any $\mathbf{s} \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$, we have $\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = m(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})\ \hat{Q}_i(\mathbf{s}, a_i)$ and $\mathbb{E}_{\mathcal{C}_i}[m(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})] = 1$ 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

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

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

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^\phi(\mathbf{s}, a_i). \end{cases} \tag{73}$$

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

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

where $\alpha_i(\mathbf{s}, a_i)\ m(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i\cup\{i\}})$ 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. □

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 $\mathcal{C}_i^k \sim Pr(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ 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 (Rashid et al., 2018). 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.72 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.