SHAQ: INCORPORATING SHAPLEY VALUE THEORY INTO MULTI-AGENT Q-LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Value factorisation proves to be a useful technique in multi-agent reinforcement learning (MARL), but the underlying mechanism is not yet fully understood. This paper explores a theoretical framework for value factorisation with interpretability. We generalise Shapley value in coalitional game theory to Markov convex game (MCG) and use it as a value factorisation method for MARL. We show that the generalised Shapley value possesses several features such as (1) efficiency: the sum of optimal generalised Shapley values is equal to the optimal global value, (2) fairness in factorisation of the global value, and (3) sensitiveness to dummy agents. Moreover, we show that MCG with the grand coalition and the generalised Shapley value is within ϵ-core, which means no agents have large incentives to deviate from the grand coalition. Since MCG with the grand coalition is equivalent to global reward game, it is the first time that Shapley value is rigorously proved to be rationally applied as a value factorisation method for global reward game. Moreover, extending from the Bellman operator we propose Shapley-Q operator that is proved to converge to the optimal generalised Shapley value. With stochastic approximation, a new MARL algorithm called Shapley Q-learning (SHAQ) is yielded. We show the performance of SHAQ on Predator-Prey for modelling relative overgeneralisation and StarCraft Multi-Agent Challenge (SMAC). In experiments, we also demonstrate the interpretability of SHAQ that is lacking in the state-of-the-art baselines.

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 search-and-rescue robots (Ramchurn et al., 2010). In this paper, we consider global reward game (also known as team reward game), i.e. an important subclass of cooperative games, where agents aim to maximize cumulative global rewards over time. There are mainly two categories of methods to solve this problem: (1) 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 (2) some scheme is applied to distribute the cumulative global rewards to each agent so that they are able to maximize the factorised value function according to their own contributions, 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 be formed as 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 viewpoint of coalitional game theory (Chalkiadakis et al., 2011), where the objective is forming coalitions and finding a value assignment scheme to distribute each coalition's value among the agents belonging to it so that no agent have large incentives to deviate from its coalition. Such a tuple involving a coalition structure (i.e., a collection of formed coalitions) and a value assignment scheme is a stable solution called core, analogous to Markov equilibrium. It is obvious that the formation of the grand coalition (i.e., only one coalition including all agents) within a core in the context of cooperative game theory naturally interprets the value factorisation in global reward game, if the value assignment scheme here is equivalently regarded as a value factorisation method.

Wang et al. (2020c) extended convex game (i.e., a type of game in coalitional game theory) (Chalkiadakis et al., 2011) to dynamic scenarios that is renamed as Markov convex game (MCG) in this paper for appropriateness. We consider MCG with the grand coalition in this paper and find a value assignment scheme so that the solution is within the ϵ-core (i.e., a relaxation of core). Although Wang et al. (2020c) gave an analytic form of Shapley value for MCG that mimics the original version (i.e., with no actions and states) for the static scenario (Shapley, 1953b), the effect of policy is neglected so that it is not promising to be within the (ϵ-)core (which means that agents are probably irrational to stay within the grand coalition with the Shapley value as the value assignment scheme). This motivates us to propose the generalised Shapley value for MCG with explicit theoretical guarantees.

Given the rationality and the other properties of the generalised Shapley value shown in this paper, we derive Shapley-Q optimality equation that is an extension of Bellman optimality equation (Bellman, 1952; Sutton & Barto, 2018). Besides, we propose Shapley-Q operator and prove its convergence to Shapley-Q optimality equation and therefore the optimal joint deterministic policy is achieved. Since Shapley-Q operator belongs to the class of value iteration algorithms, its stochastic approximation called Shapley Q-learning (SHAQ) is naturally derived. Note that the above contributions were never mentioned in the prior work.

The analytic form of the generalised Shapley value is impractical to fulfil the decentralised execution. To address this problem, with a suppose the effect of permutations on the generalised Shapley Q-value is transferred to the correlated weights $\hat{\alpha}_i(\mathbf{s}, a_i)$ and the generalised Shapley Q-value becomes fully decentralised. In implementation of SHAQ with deep reinforcement learning, to mitigate the sample inefficiency caused by permutations of agents in the prior work (Wang et al., 2020c), we force the function w.r.t. a coalition to be permutation invariant by summing up the agents' action features (i.e., the decentralised generalised Shapley Q-values), which is based on the fact that each coalition is defined as a set (Chalkiadakis et al., 2011).

To evaluate SHAQ, we run experiments on Predator-Prey for modelling the relative overgeneralisation (Böhmer et al., 2020) to show that SHAQ possesses the ability to tackle this common game-theoretic pathology (Wei & Luke, 2016) and the multi-agent StarCraft benchmark tasks (Samvelyan et al., 2019) to demonstrate the performance on the more general and challenging tasks. From the experimental results, SHAQ shows not only a generally good performance on solving all tasks, but also the interpretability that the state-of-the-art baselines lack.

2 Background

We now define Markov convex game (MCG) following the prior work (Wang et al., 2020c) (that was called extended convex game). We complement the definition with the concept of core (for dynamic scenarios rather than the original static one) that was neglected from Wang et al. (2020c).

In MCG, the equation of dynamics is defined as $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, where $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$. $\mathcal{S}$ is the set of states and $\mathcal{A} = \times_{i \in \mathcal{N}} \mathcal{A}_i$, where $\mathcal{N}$ is the set of all agents called grand coalition and $\mathcal{A}_i$ is an action set of agent i belonging to $\mathcal{N}$. If considering any coalition $\mathcal{C} \subseteq \mathcal{N}$, the joint action set of agents belonging to $\mathcal{C}$ is denoted as $\mathcal{A}_{\mathcal{C}} = \times_{i \in \mathcal{C}} \mathcal{A}_i$. $v^{\pi_{\mathcal{C}}}(\mathbf{s}) = \mathbb{E}_{\pi_{\mathcal{C}}} \big[\sum_{\tau=t}^{\infty} \gamma^{\tau - t} R_{\mathcal{C}}^{\tau} \mid \mathbf{S}_t = \mathbf{s} \big]$, where $\gamma \in (0, 1)$, represents the value function of a coalition $\mathcal{C}$ controlled by the policies of agents belonging to $\mathcal{C}$, 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 $R_{\mathcal{C}}^{\tau}$ is the reward for coalition $\mathcal{C}$ at time step τ. Accordingly, $R_{\mathcal{N}}^{\tau}$ 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 ignored for conciseness. The value of cumulative global rewards (i.e., with the grand coalition) 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 Q-value for the grand coalition is defined as $Q^{\pi}(\mathbf{s}, \mathbf{a}) \in (0, +\infty)$, and the Q-value for a coalition $\mathcal{C} \subset \mathcal{N}$ is defined as $Q^{\pi_{\mathcal{C}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}}) \in [0, +\infty)$. $Q^{\pi_{\mathcal{C}}}_{\pi_{\mathcal{D}}^*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}})$ is defined as the optimal coalitional Q-value of $\mathcal{C}$ w.r.t. the optimal joint policy of $\mathcal{D} \subseteq \mathcal{C}$, i.e. $\pi_{\mathcal{D}}^*$. $Q^{\pi_{\mathcal{C}}}_{\pi_{\mathcal{C}}^*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}})$ is denoted as $Q^{\pi_{\mathcal{C}}}_{*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}})$ for conciseness. The solution of MCG is a tuple containing a collection of coalitions called coalition structure and a value assignment scheme to agents under different coalitions. Mathematically, this tuple is represented as

$\big\langle \mathcal{CS}, \mathbf{x}(\mathbf{s}) \big\rangle$, where $\mathcal{CS} = \{\mathcal{C}_1, ..., \mathcal{C}_n\}$ is the coalition structure and $\mathbf{x}(\mathbf{s}) = \langle x_i(\mathbf{s}) \rangle_{i \in \mathcal{N}}$ is the value assignment scheme.

There is a condition for characterizing the convexity (a.k.a. supermodularity) of MCG as follows:

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

In practice, we usually 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 $v^{\pi_{\varnothing}}(\mathbf{s}) = 0$).

In MCG with the grand coalition (i.e., $\mathcal{CS} = \{\mathcal{N}\}$), core is defined as a set of value assignment schemes by which no agents have large incentives to deviate to gain more profits. Nevertheless, sometimes the exact core does not exist. To address it, we define an approximate solution called ϵ-core such that

$$\epsilon\textbf{-core} = \Big\{ \mathbf{x}(\mathbf{s}) \mid \max_{\pi_{\mathcal{C}}} x(\mathbf{s}|\mathcal{C}) \geq \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) - \epsilon, \forall \mathcal{C} \subseteq \mathcal{N}, \mathbf{s} \in \mathcal{S} \Big\}, \tag{2}$$

where $\max_{\pi_{\mathcal{C}}} x(\mathbf{s}|\mathcal{C}) = \sum_{i \in \mathcal{C}} \max_{\pi_i} x_i(\mathbf{s})$. When $\epsilon = 0$, ϵ-core becomes core. We aim to find the smallest value of ϵ that satisfy Eq.2 to obtain the least core denoted by ϵ^*-core, where $\epsilon^* = \inf\{\epsilon \mid \epsilon \text{ that makes Eq.2 hold}\}$. The definition of ($\epsilon$-)core here is a direct extension from the original definition for static convex game (Shapley, 1971; Chalkiadakis et al., 2011). The relationship between core and ϵ-core is analogous to the relationship between Nash equilibrium and epsilon equilibrium. The agent with the value assignment scheme lying in the (ϵ-)core is defined as a rational agent.

3 Generalised Shapley Value for Markov Convex Game

In this section, we (1) construct coalitional marginal contribution; (2) use coalitional marginal contribution to define the generalised Shapley value for MCG; (3) show that the generalised Shapley value lies in the ϵ^*-core of MCG with the grand coalition (i.e., rationality); and (4) show the properties of the generalised Shapley value, i.e., fairness, sensitiveness to dummy agents, and efficiency. Since the generalised Shapley value and Shapley Q-value are equivalent, we only show the theoretical results for the generalised Shapley value. Note that the proof of (3) above is important, since this gives the reason why the grand coalition exists with the generalised Shapley value as the value assignment scheme. In other words, if (3) above is invalid, then it is irrational for agents to stay within the grand coalition, since they might gain more value assignments with other coalition structures.

Coalitional Marginal Contribution. Marginal contribution has been broadly used as a credit assignment technique in the prior MARL algorithms, however, most of them were constructed based on the counterfactual regret (Foerster et al., 2018). In this section, we introduce coalitional marginal contribution by the view of coalitional game theory in Definition 1.

Definition 1. *In Markov convex game (MCG), with a sequence 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 coalitional 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 coalitional marginal contribution w.r.t. the action of each agent can be derived as follows:*

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

As Proposition 1 shows, The coalitional 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. We now clarify meaning of Eq.4. $Q_*^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i})$ indicates the optimal coalitional Q-value of $\mathcal{C}_i$ w.r.t. the optimal joint policy of $\mathcal{C}_i$. $Q_{\pi^*_{\mathcal{C}_i}}^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}})$ indicates the optimal coalitional Q-value of $\mathcal{C}_i \cup \{i\}$ w.r.t. the optimal joint policy of $\mathcal{C}_i$. Note that the agents' policies are assumed to be independent in this work.

Generalised Shapley Value. It is apparent that coalitional marginal contribution only considers one permutation to form the grand coalition. By the viewpoint from Shapley (1953b), the fairness is achieved through considering all permutations of agents. Therefore, we construct the generalised Shapley value by coalitional marginal contributions as Definition 2 shows.

Definition 2. *In MCG with the grand coalition, the generalised 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 deterministic policies, the generalised Shapley Q-value is 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 $\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i)$ is defined in Eq.4.

As Definition 2 shows, the generalised Shapley value is a linear combination of coalitional marginal contributions with different permutations of agents forming the grand coalition for promising fairness. $\frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}$ can be seen as the probability measure over $\mathcal{C}_i$, i.e., $p(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$, and the generalised Shapley value can be interpreted as the expectation of coalitional marginal contributions w.r.t. $\mathcal{C}_i \sim p(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$.

Proposition 2. *In Markov convex game with the grand coalition, the generalised Shapley value possesses several features as follows: (1) the sensitiveness to dummy agents: $v_i^{\phi}(\mathbf{s}) = 0$; (2) efficiency: $\max_{\pi} v^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} v_i^{\phi}(\mathbf{s})$.*

Proposition 2 shows 2 features of the generalised Shapley value. The most important feature is (2), which means that the maximum global value is equal to the sum of the maximum generalised Shapley values. In other words, the maximum global value is able to be reached through each agent's local optimization for its generalised Shapley value. With the fairness in Definition 2, we have showed 3 features of the generalised Shapley value.

Theorem 1. *The generalised Shapley value is a solution in the ϵ^*-core of Markov convex game (MCG) with the grand coalition and*

$$\epsilon^* = \sup_{\mathcal{C} \in \mathbb{P}(\mathcal{N}) \backslash \{\mathcal{N}, \varnothing\}} \left\{ (1 - \frac{|\mathcal{C}|!}{|\mathcal{N}|!}) \cdot \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) \right\}.$$

$\mathbb{P}(\cdot)$ denotes the power set. Theorem 1 shows the rationality of the generalised Shapley value, i.e., no agents have large incentives to deviate from the grand coalition. In other words, the equivalence between MCG with the grand coalition and global reward game almost holds, with the generalised Shapley value proposed in this paper as a value assignment scheme. As a result, the generalised Shapley value is reasonable to be a value factorisation method in global reward game.

4 Shapley Q-Learning

In this section, we firstly derive the Shapley-Q optimality equation by incorporating the efficiency of the generalised Shapley Q-value into the Bellman optimality equation for the global Q-value, given the results in Theorem 1 that no agents have large incentives to deviate from the grand coalition. Then, we propose the Shapley Q-operator that can converge to the optimal generalised Shapley Q-value and therefore the optimal joint deterministic policy can be achieved. For easy implementation, we derive the Shapley Q-learning and its variant for fitting decentralised execution. The construction of modules for Shapley Q-learning follows the convergence condition in Theorem 2, so that the convergence of the learning process is possible to be guaranteed.

Shapley-Q Optimality Equation. Based on Bellman optimality equation (Bellman, 1952; Sutton & Barto, 2018), we derive an equation called Shapley-Q optimality equation for evaluating the optimal generalised Shapley value (see Appendix B.4) such that

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

where $\mathbf{I}$ is an identity matrix; $\mathbf{w}(\mathbf{s},\mathbf{a}) = [w_i(\mathbf{s},a_i)]^\top \in \mathbb{R}_+^{|\mathcal{N}|}$; $\mathbf{Q}^{\phi^*}(\mathbf{s},\mathbf{a}) = [Q_i^{\phi^*}(\mathbf{s},a_i)]^\top \in \mathbb{R}_+^{|\mathcal{N}|}$ and $Q_i^{\phi^*}(\mathbf{s},a_i)$ denotes the optimal generalised Shapley Q-value. Given the theoretical results in Theorem 1 that almost no agents have large incentives to deviate from the grand coalition, the derivation of Eq.7 also depends on two conditions: (1) the efficiency of the generalised Shapley value proved in Proposition 2; (2) a suppose that $Q_i^{\phi^*}(\mathbf{s},a_i) = w_i(\mathbf{s},a_i)\ Q_*^\pi(\mathbf{s},\mathbf{a})$, where $Q_*^\pi(\mathbf{s},\mathbf{a}) > 0$. 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 $w_i(\mathbf{s},a_i^*) = 1/|\mathcal{N}|$. Literally, the credit assignments would be equal and each agent would receive $Q_*^\pi(\mathbf{s},\mathbf{a})/|\mathcal{N}|$ if making optimal decisions. It is apparent that the efficiency holds under this situation and this can be interpreted as an extremely fair credit assignment, i.e., the award 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. Note that the value of $w_i(\mathbf{s},a_i)$ for $a_i \neq \arg\max_{a_i} Q_i^{\phi^*}(\mathbf{s},a_i)$ is needed to find (e.g., during learning).

Shapley-Q Operator. To find an optimal solution given by Eq.7, we now propose an operator called Shapley-Q 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:

$$\Upsilon\big(\times_{i\in\mathcal{N}} Q_i^{\phi}(\mathbf{s},a_i)\big) = \mathbf{I}\,\mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a})\big[R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}',a_i)\big], \tag{8}$$

where $w_i(\mathbf{s},a_i) = 1/|\mathcal{N}|$ when $a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s},a_i)$. We prove that the optimal joint deterministic policy can be achieved through the Shapley-Q operator in Theorem 2.

Theorem 2. *Shapley-Q operator can converge to the optimal Shapley Q-values and therefore the optimal joint deterministic policy is achieved 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 derive the TD error for Shapley Q-learning (SHAQ) based on Shapley-Q operator (see Appendix B.4) such that

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

where

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

Here $\alpha_i(\mathbf{s},a_i) = \frac{1}{|\mathcal{N}|\ w_i(\mathbf{s},a_i)}$ for $a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s},a_i)$. In implementation, we should guarantee the condition stated in Theorem 2 for the convergence of SHAQ to the optimality.

Implementation of Shapley Q-Learning. We now describe a practical implementation of SHAQ for Dec-POMDP (Oliehoek, 2012), i.e. global reward game but with partial observations during execution. For this reason, the history of each agent is substituted for the global state of each generalised Shapley Q-value in implementation to guarantee the optimal deterministic joint policy (Oliehoek, 2012). Generalised Shapley Q-value is now replaced by $Q_i^{\phi}(\tau_i,a_i)$, where τ_i is a history of partial observations of agent i. Since centralised training decentralised execution (CTDE) (Oliehoek et al., 2008) is applied, the global state s for $\hat{\alpha}_i(\mathbf{s},a_i)$ is able to be obtained during training.

Proposition 3. *Suppose any coalitional 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*

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

we have

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

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

Compatible with 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 3, the information of coalition can be equivalently transferred to $\hat{\alpha}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})$. As a result, $\delta_i(\mathbf{s}, a_i)$ is equivalently transferred to the form as follows:

$$\hat{\delta}_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \\ \hat{\alpha}_i(\mathbf{s}, a_i) = \mathbb{E}_{\mathcal{C}_i \sim p(\mathcal{C}_i | \mathcal{N} \backslash \{i\})}[\ \hat{\alpha}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})\ ] & a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i). \end{cases} \tag{12}$$

To solve partial observations, $\hat{Q}_i(\tau_i, a_i)$ is represented as recurrent neural network (RNN) with GRUs (Chung et al., 2014). $\hat{\alpha}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})$ is approximated by a parametric function $F_{\mathbf{s}} + 1$ such that

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

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 $\mathcal{C}_i^k \sim p(\mathcal{C}_i | \mathcal{N} \backslash \{i\})$ is sampled M times to approximate $\mathbb{E}_{\mathcal{C}_i \sim p(\mathcal{C}_i | \mathcal{N} \backslash \{i\})}[\ \hat{\alpha}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})\ ]$; 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 in Theorem 2 (see Appendix B.5) such that $\max_{\mathbf{s}} \big\{ \sum_{i \in \mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \big\} < \frac{1}{\gamma}$.

By using the framework of fitted Q-learning (Ernst et al., 2005) and inserting the above constructed variables, the practical loss function derived from Eq.9 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 C.

5 Related Work

Value Factorisation. 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 learned a global Q-value that can be regarded as identically assigning to all agents with same credits during training (Wang et al., 2020c), which may cause the unfair value assignment and therefore the difficulty for learning (Wolpert & Tumer, 2002). To avoid this problem, VDN (Sunehag et al., 2018) was proposed to automatically 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 problem, 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 mainly for showing the convergence to the optimal joint deterministic policy. Shapley Q-value proposed in this paper belongs to the family of value factorisation methods, but based on the theoretical framework of MCG that is interpretable in theory and with the realistic insights from static 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). These games show meaningful representations for coalition and its reward, which verifies the realistic existence of the concepts introduced in MCG. In real world, these concepts cannot be always explicitly defined, however, it can be assumed that these concepts inherently exist and only the exposed concept (e.g. the global reward) needs to be focused on. Compared to IGM, the efficiency of the generalised Shapley Q-value plays a significant role of proving the convergence to the optimal joint deterministic policy. The generalised Shapley value can be categorised as a sort of linear value factorisation and the efficiency of linear value factorisation was also studied by Wang et al. (2020a) from another perspective of analysis.

Relationship to VDN. It is apparent to observe that by setting $\delta_i(\mathbf{s}, a_i) = 1$ for any state-action pairs, SHAQ is degraded to VDN (Sunehag et al., 2018). In other words, VDN is a subclass of SHAQ.

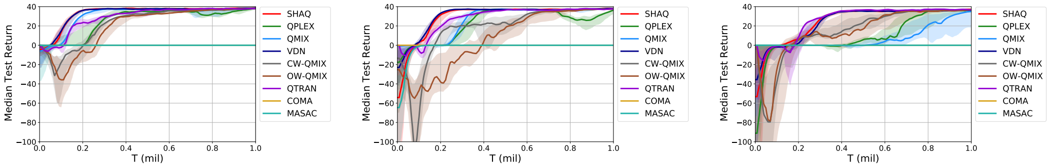
As a result, we show the reason why VDN works well in most scenarios by the framework of MCG proposed in this paper. In this sense, its poor performance on some scenarios is due to the mistakes on defining $\delta_i(\mathbf{s}, a_i) = 1$ in Eq.9 over the sub-optimal actions.

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. (1) As Wang et al. (2020c) claimed, there exists an efficient value assignment schemes that guarantees a solution in the core of MCG with the grand coalition. We show that this value assignment scheme is actually coalitional marginal contribution proposed in this paper (see Proposition 5 in Appendix B.2). The fault led by Wang et al. (2020c) is that the property of MCG is misunderstood as identical to that of the original convex game (Chalkiadakis et al., 2011). However, in our results the consideration of policy changes the property of MCG and lead to the different theoretical results for Shapley value (see Theorem 1). Fortunately, the correctness of functional approximation for Shapley Q-value by Wang et al. (2020c) is unaffected, and it can be regarded as an implementation of the generalised Shapley Q-value proposed in this paper. (2) We propose a learning algorithm called Shapley Q-learning that was never mentioned in the prior work. (3) In implementation, the sample complexity of SQDDPG for different permutations of agents is higher than that of SHAQ. For discrete states and actions, SQDDPG attempts to directly learn the value of different permutations (i.e., $|\mathcal{S}||\mathcal{A}||\mathcal{N}|!$ possible Q-values), while SHAQ applies the fact that the value for the same coalition is identical (since a coalition is a set) and learns an invariant value function for each coalition (i.e., $|\mathcal{S}||\mathcal{A}|\big[\ |\mathcal{N}|! - \sum_{\mathcal{C}\subseteq\mathcal{N}}[\ |\mathcal{C}|!(|\mathcal{N}| - |\mathcal{C}|)! - 1\ ]\ \big]$ possible Q-values).

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) [1]. 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 D.1, whereas the implementation of baselines are from Rashid et al. (2020) [2]. We also compare SHAQ with SQDDPG (Wang et al., 2020c) [3], which is left to Appendix E.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 easy and hard maps in SMAC, we apply 50k time steps for annealing, the same as that in Samvelyan et al. (2019); while for 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 5000 for all algorithms that is the same as 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 for Modelling Relative Overgeneralisation



(a) p = - 0.5.

(b) p = - 1.

(c) p = - 2.

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

[1] 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.

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

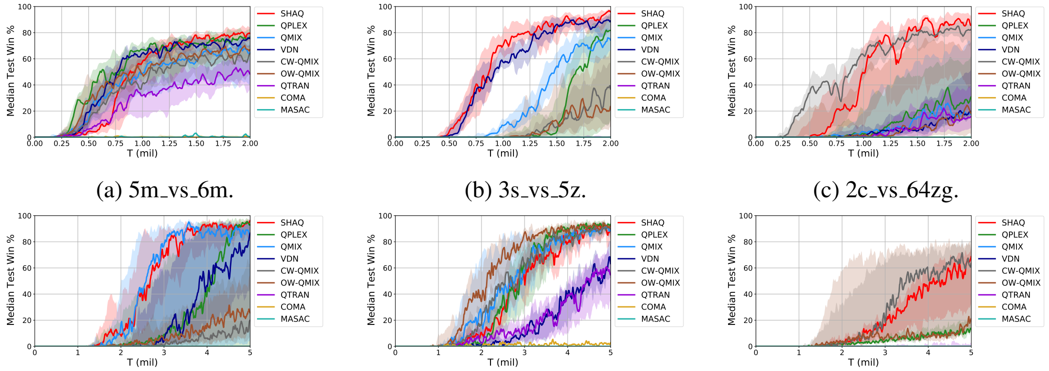
[3] The code of SQDDPG is implemented based on `https://github.com/hsvgbkhgbv/SQDDPG`.

In this section, we run the experiments on a partially-observable task called Predator-Prey for modelling relative overgeneralisation (Böhmer et al., 2020). Relative overgeneralisation is a common game theoretic pathology that the suboptimal actions is preferred when matched with arbitrary actions from the collaborating agents (Wei & Luke, 2016), where 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. 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. For this reason, we apply the identical epsilon annealing schedule (i.e. 1 million time steps) with that in Wang et al. (2020b). As Figure 1 shows, SHAQ can solve the relative overgeneralisation tasks with different levels (i.e., 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. 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 failures of MASAC and COMA could be due to that relative obvergeneralisation 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 of results in Appendix E.2.

Performance Analysis. First, we compare performances between SHAQ and baselines. It shows in Figure 2 that SHAQ outperforms all baselines on 3 hard maps, meanwhile SHAQ can tackle the challenging super-hard maps. For 3s5z_vs_3s6z, SHAQ possesses the fastest convergence rate and the best final performance. For MMM2, SHAQ suffers from the slower convergence rate but can achieve a good final performance. On 6h_vs_8z, SHAQ can beat other baselines except for CW-QMIX. Surprisingly, VDN performs well on most of hard maps, which verifies our analysis in Section 5. QMIX and QPLEX perform well on the most of maps, except for 3z_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 in challenging tasks. Although QTRAN can theoretically represent the complete class of global Q-value (Son et al., 2019), its complicated learning paradigm could impede its generalisation to challenging tasks which results in its poor performance. W-QMIX can perform better than QMIX on some maps that verifies that the problem for QMIX on the restricted representation of global Q-value is fixed to some extent. Nevertheless, due to the inevitable hyperparameter tuning with no laws Rashid et al. (2020), it is difficult to be generalised to all scenarios (see Appendix E.3).



(a) 5m_vs_6m.

(b) 3s_vs_5z.

(c) 2c_vs_64zg.

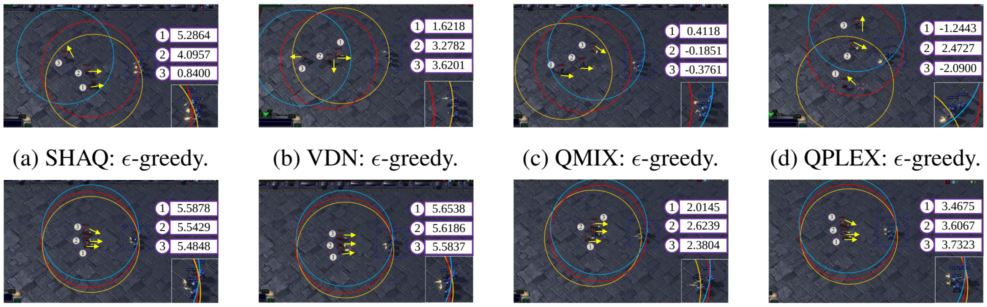
(d) 3s5z_vs_3s6z.

(e) MMM2.

(f) 6h_vs_8z.

Figure 2: Median test win % for hard (a-c), and super-hard (d-f) maps of SMAC.

Interpretability of SHAQ. To show the interpretability of SHAQ, we conduct a test on 3m (i.e. a simple task in SMAC) with both ϵ-greedy policy (for obtaining both optimal and sub-optimal decisions) and greedy policy (for certainly obtaining optimal decisions). As seen from Figure 3a, Agent 3 faces the direction opposite to enemies, meanwhile, the enemies are apparently out of its attacking range. It means that Agent 3 does not contribute to the team, i.e., it is almost a dummy agent, so it only receives 0.8400 (near 0) as its generalised Shapley value. In contrast, Agent 1 and Agent 2 are attacking enemies. However, Agent 1 additionally suffers from more attacks (with lower health) than Agent 2. As a result, Agent 1 contributes more than Agent 2 and therefore receives more credits as its generalised Shapley value. On the other hand, we can see from Figure 3e that with the optimal policies all agents receive almost identical credits as generalised Shapley values. This is consistent with the analysis of Shapley-Q optimality equation that agents should receive equal credits if they execute the optimal joint actions. Note that if any agent executes a sub-optimal action, it will earn different credits that is deserved. These all together constructs the fairness of credit assignments. In summary, we demonstrate that (1) when agents perform optimal policies, they receive almost identical credits as generalised Shapley values; (2) when an agent performs as a dummy agent, it receives credits near 0 as its generalised Shapley value; and (3) each agent's generalised Shapley value is positively correlated to its contribution, so in fairness. To verify that the generalised Shapley values learned by SHAQ is non-trivial, we also show the results for 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. Since VDN is a subclass of SHAQ and possesses the same form of loss function for optimal actions, it is reasonable to obtain the similar results to SHAQ. The explanation for results of QMIX and QPLEX deserves to be studied in the future work. As for the sub-optimal decisions, VDN does not possess an explicit interpretation as SHAQ due to incorrectly defining $\delta_i(\mathbf{s}, a_i) = 1$ over sub-optimal actions, which verifies our analysis in Section 5. QMIX and QPLEX cannot show explicit interpretations over sub-optimal decisions.



(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 3: Visualisation of the evaluation 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.

7 Conclusion

This paper generalises Shapley value in coalitional game theory to Markov Convex Game (MCG) through constructing an appropriate analytic form of coalitional marginal contribution. We also formally prove the rationality of Shapley value for MCG with the grand coalition (i.e. equivalent to global reward game), so that it can be used as a value factorisation method for global reward game. Moreover, we show the property of the generalised Shapley value: (1) efficiency, (2) sensitiveness to dummy agents, and (3) fairness. By the property of efficiency, we propose Shapley-Q operator with the proof of convergence to the optimal joint deterministic policy as well as its stochastic approximation called Shapley Q-learning (SHAQ). We evaluate SHAQ on Predator-Prey for modelling relative overgeneralisation (Böhmer et al., 2020) and the challenging multi-agent StarCraft benchmark tasks (Samvelyan et al., 2019). SHAQ shows generally good performance with interpretability, compared with state-of-the-art multi-agent reinforcement learning algorithms. Specifically, it verifies the theoretical results such as sensitiveness to dummy agents and fairness via complicated experiments.

ETHICS STATEMENT

This work is still standing on the theoretical analysis, so no further social impacts or ethical problems will be caused at the moment.

REPRODUCIBILITY STATEMENT

In this section, we discuss the reproducibility of this work. First, the complete proofs of the theoretical results are shown in Appendix. Specifically, the assumptions and some preliminary theoretical results that aids the proofs of main theoretical claims are shown in Appendix B.1 and B.2 respectively. The proofs of theoretical claims in Section 3 are shown in Appendix B.3. The proofs of theoretical claims and derivations in Section 4 are shown in Appendix B.4 and B.5. To help readers understand the theoretical framework defined in Section 2, we also provide specific discussions and examples in Appendix A.2. The details of experiments are shown in Section 6 and Appendix D. The pseudo code of Shapley Q-learning (SHAQ) is shown in Appendix C, for helping implementing the algorithm. Finally, we also include the source code for implementing SHAQ in the supplementary materials.

REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks. 2017. URL `https://openreview.net/pdf?id=rkpACe1lx`.

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

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

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

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176. PMLR, 2019.

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

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

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

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

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

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Anuj Mahajan, Mikayel Samvelyan, Lei Mao, Viktor Makoviychuk, Animesh Garg, Jean Kossaifi, Shimon Whiteson, Yuke Zhu, and Animashree Anandkumar. Tesseract: Tensorised actors for multi-agent reinforcement learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7301–7312. PMLR, 2021.

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

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

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

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

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

Sarvapali D Ramchurn, Alessandro Farinelli, Kathryn S Macarthur, and Nicholas R Jennings. Decentralized coordination in robocup rescue. *The Computer Journal*, 53(9):1447–1461, 2010.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. RODE: learning roles to decompose multi-agent tasks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

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

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

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

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

A Additional Background

A.1 Value Factorisation in MARL

Although there are lots of works on (Q-)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*

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

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

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

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

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 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

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

where

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

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.

A.2 Interpretation of Definitions for MCG

Condition of Markov Convex Game. Eq.1 implies a fact existing in most real-life scenarios that a larger coalition results in the greater optimal global value in cooperation and therefore greater optimal value (credit) assignments (see Remark 1) that directly increases the agents' incentives for joining the grand coalition. This can be regarded as a reason for raising and studying global reward game with value factorisation. This interpretation for dynamic scenarios in this paper is consistent with the one given by Shapley (1971) for the static scenarios, known as the snowball effect.

Remark 1. *Suppose that $\mathcal{C}_1 = \{1,2\}$ and $\mathcal{C}_2 = \{3\}$, so $\mathcal{C}_{\cup} = \mathcal{C}_1 \cup \mathcal{C}_2 = \{1,2,3\}$ and $\mathcal{C}_{\cap} = \mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$. Therefore, by Eq.1 we can write that $\max_{\pi_{\{1,2,3\}}} v^{\pi_{\{1,2,3\}}}(\mathbf{s}) \geq \max_{\pi_{\{1,2\}}} v^{\pi_{\{1,2\}}}(\mathbf{s}) + \max_{\pi_{\{3\}}} v^{\pi_{\{3\}}}(\mathbf{s})$. As a result, we can get that*

$$\max_{\pi_{\{1,2,3\}}} v^{\pi_{\{1,2,3\}}}(\mathbf{s}) - \max_{\pi_{\{1,2\}}} v^{\pi_{\{1,2\}}}(\mathbf{s}) = \max_{\pi_{\{3\}}} \Phi_{\{3\}}(\mathbf{s}|\mathcal{C}_{\{1,2\}}) \geq \max_{\pi_{\{3\}}} v^{\pi_{\{3\}}}(\mathbf{s}). \tag{19}$$

Since $\mathcal{C}_2$ only includes agent 3, $\max_{\pi_{\{3\}}} v^{\pi_{\{3\}}}(\mathbf{s})$ is naturally agent 3's optimal value assignment here. It is obvious that with a larger coalition, agent 3's optimal value assignment increases.

Insight into ϵ-core. In Eq.2, $\mathbf{x}(\mathbf{s}) = \langle x_i(\mathbf{s}) \rangle_{i \in \mathcal{N}}$ indicates the value assignment scheme for the grand coalition. In the context of cooperative game theory, it is called imputation vector. $\max_{\pi_{\mathcal{C}}} x(\mathbf{s}|\mathcal{C}) = \sum_{i \in \mathcal{C}} \max_{\pi_i} x_i(\mathbf{s})$ indicates the sum of value assignments (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 optimal global value obtained by the value assignment scheme in the ϵ-core with the grand coalition is almost 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 ϵ-core (with the grand coalition).

Remark 2. *Suppose that a coalition structure is $\mathcal{CS} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n\}$ 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 ϵ-core holds for the grand coalition (i.e. $\mathcal{N}$) with some value assignment scheme $\mathbf{x}(\mathbf{s}) = \langle x_i(\mathbf{s}) \rangle_{i \in \mathcal{N}}$. 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*

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

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

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

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:

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

where $\max_{\pi} \hat{v}^{\pi}(\mathbf{s})$ is denoted as the optimal global value obtained by the value assignment scheme in the ϵ-core. By the result in Remark 2, the RHS of Eq.21 can be written as follows:

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

where $\max_{\pi} v^{\pi}(\mathbf{s})$ is the optimal global value obtained by coalition structures. By substituting Eq.22 and 23 into Eq.21, we can get that

$$\max_{\pi} \hat{v}^{\pi}(\mathbf{s}) \geq \max_{\pi} v^{\pi}(\mathbf{s}) - |\mathcal{CS}|\epsilon.$$

If $|\mathcal{CS}|\epsilon \geq 0$ is bounded, we say that the optimal global value obtained by the value assignment scheme in the ϵ-core is almost no less than the optimal global value obtained by $\mathcal{CS}$. Note that the value of $|\mathcal{CS}|\epsilon$ should be as small as possible.

B Complete Mathematical Proofs

B.1 Assumptions

Assumption 1. *Any joint policy can be factorised to a permutation of decentralised (i.e. disjoint) policies based on predecessor coalitions, i.e., $\pi_{\mathcal{C}} = \times_{\mathcal{C}_i \in \Pi(\mathcal{C})} \pi_i(\mathcal{C}_i)$. $\Pi(\mathcal{C})$ here is a set of predecessor coalitions induced by an arbitrary permutation (or a sequence) of agents to form a coalition $\mathcal{C}$ and $\pi_i(\mathcal{C}_i)$ is a policy of agent i for the predecessor coalition $\mathcal{C}_i$. This is consistent with the definition of $v^{\pi_{\mathcal{C}}}(\mathbf{s})$ that is a set-valued function. If only one permutation is considered, $\pi_i(\mathcal{C}_i)$ will be denoted as π_i for simplification and conciseness.*

Assumption 2. *The functional space of each agent i's policy π_i is able to be separable with respect to the predecessor coalitions, i.e., $\pi_i = \biguplus_{\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}} \pi_i(\mathcal{C}_i), \ \forall i \in \mathcal{N}$, where $\biguplus$ denotes the disjoint union. For example, $\pi_i(\mathcal{C}_i) \cap \pi_i(\mathcal{D}_i) = \varnothing$, if $\mathcal{C}_i \neq \mathcal{D}_i, \ \forall \mathcal{C}_i, \mathcal{D}_i \in \mathcal{N} \backslash \{i\}$. 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.*

Assumption 3. *If an agent $i \in \mathcal{N}$ is dummy it will not give any contribution to any predecessor coalition $\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}$, meanwhile, no members in the predecessor coalition $\mathcal{C}_i$ will react in different manners after agent i joins.*

B.2 Preliminary Theoretical Results

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

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

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

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

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

Therefore, we can get the equation such that

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

With the same conditions, the reverse direction of proof apparently holds by going through from Eq.26 to 24. □

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

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

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

where $\max_{\pi_{\mathcal{C}_i \cup \{i\}}} v^{\pi_{\mathcal{C}_i}}(\mathrm{s}) = \max_{\pi_{\mathcal{C}_i}} v^{\pi_{\mathcal{C}_i}}(\mathrm{s})$, 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 $v^{\pi_\varnothing}(\mathrm{s}) = 0$ and the result from Eq.27, by Assumption 1 we can get the equations such that

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

□

Theorem 3. *Coalitional marginal contribution is a solution in the 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 coalitional marginal contribution is a value assignment scheme in the 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

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

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

Suppose for the sake of contradiction that we have $\max_{\pi_{\mathcal{C}}} \Phi(\mathrm{s}|\mathcal{C}) < \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathrm{s})$ for some $\mathrm{s} \in \mathcal{S}$ 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 sequence $\langle j_1, j_2, ..., j_{|\mathcal{C}|} \rangle$, 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:

$$\begin{aligned} \max_{\pi_\mathcal{C}} v^{\pi_\mathcal{C}}(\mathbf{s}) = &\max_{\pi_{j_1}} v^{\pi_{j_1}}(\mathbf{s}) - \max_{\pi_\emptyset} v^{\pi_\emptyset}(\mathbf{s}) \\ &+ \max_{\pi_{\{j_1, j_2\}}} v^{\pi_{\{j_1, j_2\}}}(\mathbf{s}) - \max_{\pi_{j_1}} v^{\pi_{j_1}}(\mathbf{s}) \\ &+ \qquad \vdots \\ &+ \max_{\pi_\mathcal{C}} v^{\pi_\mathcal{C}}(\mathbf{s}) - \max_{\pi_{\mathcal{C} \backslash \{j_n\}}} v^{\pi_{\mathcal{C} \backslash \{j_n\}}}(\mathbf{s}), \end{aligned}$$

and due to Definition 1 we can obviously get the following equations such that

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

$$\Downarrow$$

$$\max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) = \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}^n_k) = \max_{\pi_{\mathcal{C}^n_\cup}} v^{\pi_{\mathcal{C}^n_\cup}}(\mathbf{s}) - \max_{\pi_{\mathcal{C}^n_k}} v^{\pi_{\mathcal{C}^n_k}}(\mathbf{s}),$$

by adding up these inequalities in Eq.31 for all $\mathcal{C} \in \mathcal{N}$, we can directly obtain a new inequality such that

$$\sum_{i \in \mathcal{C}} \max_{\pi_i} \Phi_i(\mathbf{s}|\mathcal{C}_i) = \max_{\pi_\mathcal{C}} \Phi(\mathbf{s}|\mathcal{C}) \geq \max_{\pi_\mathcal{C}} v^{\pi_\mathcal{C}}(\mathbf{s}). \tag{32}$$

It is obvious that Eq.32 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 coalitional marginal contribution is a solution in the core of Markov convex game (MCG) with the grand coalition. □

B.3 Mathematical Proofs for the Generalised Shapley Value

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

$$\Phi_i(\mathbf{s}, a_i|\mathcal{C}_i) = \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}_{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}_{*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}). \tag{33}$$

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:

$$\begin{aligned} \max_{\pi_{\mathcal{C}_i}} v^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}) &= \max_{\pi_{\mathcal{C}_i}} \sum_{\mathbf{a}_{\mathcal{C}_i \cup \{i\}}} \pi_{\mathcal{C}_i \cup \{i\}}(\mathbf{a}_{\mathcal{C}_i \cup \{i\}}|\mathbf{s}) \ Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \\ &\quad \Big(\text{Since } \pi_{\mathcal{C}_i \cup \{i\}} \text{ is a deterministic joint policy, we can have the following equation.}\Big) \\ &= \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \\ &\quad \Big(\text{ We write } \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \text{ as } Q^{\pi_{\mathcal{C}_i \cup \{i\}}}_{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) \Big) \\ &= \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}_{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}). \end{aligned} \tag{34}$$

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

$$\max_{\pi_{\mathcal{C}_i}} v^{\pi_{\mathcal{C}_i}}(\mathbf{s}) = \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) = \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}_{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) = \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}_{*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}). \tag{35}$$

Since $\max_{\pi_{\mathcal{C}_i}} v^{\pi_{\mathcal{C}_i}}(\mathbf{s})$ is irrelevant to a_i, by Eq.34 and 35 we can get that

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

By Eq.36, we can get the following result such that

$$\begin{aligned}
\Phi_i^*(\mathbf{s}, a_i | \mathcal{C}_i) &= \max_{\pi_i} \Phi_i(\mathbf{s}, a_i | \mathcal{C}_i) \\
&= \max_{\pi_i} \left\{ \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}_{\pi^*_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}_{*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \right\} \\
&= \max_{\pi_i} \left\{ \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \right\} \\
&= \max_{\pi_i} \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \\
&= \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i \cup \{i\}}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} \max_{\pi_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}) \\
&= \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i \cup \{i\}}}_{*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i \cup \{i\}}) - \max_{\mathbf{a}_{\mathcal{C}_i}} Q^{\pi_{\mathcal{C}_i}}_{*}(\mathbf{s}, \mathbf{a}_{\mathcal{C}_i}).
\end{aligned} \tag{37}$$

The proof is completed. □

Lemma 1. *For any agent $i \in \mathcal{N}$, $\forall \mathbf{s} \in \mathcal{S}$, its optimal generalised Shapley value denoted as $\max_{\pi_i} v_i^{\phi}(\mathbf{s})$ satisfies the following equation such that*

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

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

$$\max_{\pi_i} v_i^{\phi}(\mathbf{s}) \leq \sum_{\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i} \Phi_i(\mathbf{s} | \mathcal{C}_i). \tag{38}$$

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 2 claims, we can write Eq.38 as follows:

$$\max_{\pi_i} v_i^{\phi}(\mathbf{s}) = \sum_{\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s} | \mathcal{C}_i), \tag{39}$$

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

Remark 4. *Usually, it is difficult to make decisions in parallel for all permutations to form the grand coalition. Instead, it learns one π_i (i.e. a policy projection) that can approximate (or take place of) all $\pi_i(\mathcal{C}_i)$ with respect to $\mathcal{C}_i \subseteq \mathcal{N}$, minimising the distance that is defined as follows:*

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

As Eq.38 shows, $\max_{\pi_i} v_i^{\phi}(\mathbf{s})$ is a lower bound of $\sum_{\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i} \Phi_i(\mathbf{s} | \mathcal{C}_i)$, so the distance above perhaps never converges to 0 in practice, i.e., there exists an error bound. If the error bound ϵ such that

$$\max_{\pi_i} \Big| \max_{\pi_i} v_i^{\phi}(\mathbf{s}) - \sum_{\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i} \Phi_i(\mathbf{s} | \mathcal{C}_i) \Big| \leq \epsilon$$

is extremely small such that $\epsilon \to 0$, there should be almost no gap between the theoretical results analysed in this paper and the practical results. Nevertheless, in theory we can directly use Eq.39 for the convenience of analysis.

Proposition 2. *In Markov convex game with the grand coalition, the generalised Shapley value possesses several features as follows: (1) the sensitiveness to dummy agents: $v_i^{\phi}(\mathbf{s}) = 0$; (2) efficiency: $\max_{\pi} v^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} v_i^{\phi}(\mathbf{s})$.*

Proof. The complete proof is as follows. We will firstly prove the (1), then (2). For any agent $i \in \mathcal{N}$, $\forall \mathbf{s} \in \mathcal{S}$, its generalised Shapley value denoted as $v_i^{\phi}(\mathbf{s})$.

Proof of (1). 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 3, for any predecessor coalition $\mathcal{C}_i^m \subseteq \mathcal{N} \backslash \{i\}$ we have $\max_{\pi_{\mathcal{C}_i^m}} v^{\pi_{\mathcal{C}_i^m}}(\mathbf{s}) = \max_{\pi_{\mathcal{C}_i^m}} v^{\pi_{\mathcal{C}_i^m \cup \{i\}}}(\mathbf{s}),\ \forall \mathbf{s} \in \mathcal{S}$, thereby $\Phi_i(\mathbf{s}|\mathcal{C}_i^m) = 0$. Also, the above analysis fulfills for all permutations of agents to form the grand coalition. By the definition of the generalised Shapley value in Markov convex game shown in Definition 2, it is not difficult to see that the generalised Shapley value for the dummy agent will be 0 such that $v_i^{\phi}(\mathbf{s}) = 0$. The proof of (1) completes.

Proof of (2). The objective is proving that the generalised Shapley value satisfies the following equation such that

$$\max_{\pi} v^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i} v_i^{\phi}(\mathbf{s}),\ \forall \mathbf{s} \in \mathcal{S}, \tag{40}$$

where $v_i^{\phi}(\mathbf{s})$ denotes the generalised Shapley value. By the result from Proposition 5 and Assumption 1, for an arbitrary sequence $m \in \Pi(\mathcal{N})$ we can get the equation such that

$$\max_{\pi} v^{\pi}(\mathbf{s}) = \sum_{i \in \mathcal{N}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m),\ \forall \mathbf{s} \in \mathcal{S}, \tag{41}$$

where $\Phi_i(\mathbf{s}|\mathcal{C}_i^m)$ is a coalitional marginal contribution and $\pi_i(\mathcal{C}_i^m)$ ($\mathcal{C}_i^m$ is the predecessor coalition that agent i meets in the sequence m) is a sub-policy of agent $i \in \mathcal{N}$ for the sequence m. If we consider all possible sequences of agents to form the grand coalition and add all these inequalities, we can get the following equation such that

$$\begin{gathered}
\sum_{m \in \Pi(\mathcal{N})} \max_{\pi} v^{\pi}(\mathbf{s}) = \sum_{m \in \Pi(\mathcal{N})} \sum_{i \in \mathcal{N}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m),\ \forall \mathbf{s} \in \mathcal{S}. \\
\Downarrow \\
\frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi} v^{\pi}(\mathbf{s}) = \frac{1}{|\mathcal{N}|!} \sum_{i \in \mathcal{N}} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m),\ \forall \mathbf{s} \in \mathcal{S}.
\end{gathered} \tag{42}$$

Next, to ease life we start from the left hand side of Eq.42. We can directly get the following equation such that

$$\frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi} v^{\pi}(\mathbf{s}) = \frac{1}{|\mathcal{N}|!} \cdot |\mathcal{N}|! \cdot \max_{\pi} v^{\pi}(\mathbf{s}|\mathcal{N}) = \max_{\pi} v^{\pi}(\mathbf{s}|\mathcal{N}). \tag{43}$$

Now, we start processing the right hand side of Eq.42. By rearranging it, we can get the equations such that

$$\begin{aligned}
\frac{1}{|\mathcal{N}|!} \sum_{i \in \mathcal{N}} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) &= \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|!} \sum_{m \in \Pi(\mathcal{N})} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) \\
&\quad \text{(The identical } \mathcal{C}_i^m \text{ in different sequences is written as } \mathcal{C}_i \\
&\quad \text{ and we can rearrange the equation as follows.)} \\
&= \sum_{i \in \mathcal{C}} \frac{1}{|\mathcal{N}|!} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}} |\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)! \cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i) \\
&= \sum_{i \in \mathcal{N}} \sum_{\mathcal{C}_i \subseteq \mathcal{N} \backslash \{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}| - |\mathcal{C}_i| - 1)!}{|\mathcal{N}|!} \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i).
\end{aligned} \tag{44}$$

By Lemma 1, we can get the following equations such that

$$\sum_{i\in\mathcal{N}} \sum_{\mathcal{C}_i\subseteq\mathcal{N}\backslash\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!} \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i) \tag{45}$$

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

$$= \sum_{i\in\mathcal{N}} \max_{\pi_i} v_i^{\phi}(\mathbf{s}). \tag{46}$$

Inserting the results from Eq.43 and 46 to Eq.42, we can get the equation such that

$$\max_{\pi} v^{\pi}(\mathbf{s}|\mathcal{N}) = \sum_{i\in\mathcal{N}} \max_{\pi_i} v_i^{\phi}(\mathbf{s}),\ \forall \mathbf{s}\in\mathcal{S}. \tag{47}$$

Therefore, the proof for (2) completes. □

Theorem 1. *The generalised Shapley value is a solution in the ϵ^*-core of Markov convex game (MCG) with the grand coalition and*

$$\epsilon^* = \sup_{\mathcal{C}\in\mathbb{P}(\mathcal{N})\backslash\{\mathcal{N},\varnothing\}} \left\{(1-\frac{|\mathcal{C}|!}{|\mathcal{N}|!})\cdot \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s})\right\}.$$

Proof. The complete proof is as follows.

Proof sketch. We need to construct the inequality similar to that in the proof of Theorem 3. Different from the proof in Theorem 3, we should consider all permutations of sequences generated from $\{j_1, j_2, ..., j_{|\mathcal{C}|}\}$ rather than an arbitrary sequence. For the result with an arbitrary sequence from the proof of Theorem 3, we can directly apply it to the proof here.

Similar to the aim of proof in Theorem 3, we just need to show that for any intermediate coalition $\mathcal{C}\subseteq\mathcal{N}$ the following condition should be satisfied such that

$$\max_{\pi_{\mathcal{C}}} \tilde{x}(\mathbf{s}|\mathcal{C}) \geq \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}),\ \forall \mathbf{s}\in\mathcal{S}, \tag{48}$$

where $\max_{\pi_{\mathcal{C}}} \tilde{x}(\mathbf{s}|\mathcal{C}) = \sum_{i\in\mathcal{C}} \max_{\pi_i} v_i^{\phi}(\mathbf{s})$ and $v_i^{\phi}(\mathbf{s})$ is denoted as the Shapley value of an agent belonging to the coalition $\mathcal{C}$. As Eq.5 shows, $v_i^{\phi}(\mathbf{s})$ can be expressed as the equation as follows:

$$v_i^{\phi}(\mathbf{s}) = \sum_{\mathcal{C}_i\subseteq\mathcal{N}\backslash\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!} \cdot \Phi_i(\mathbf{s}|\mathcal{C}_i). \tag{49}$$

By Lemma 1, we can get the equation such that

$$\max_{\pi_i} v_i^{\phi}(\mathbf{s}) = \sum_{\mathcal{C}_i\subseteq\mathcal{N}\backslash\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!} \cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i), \tag{50}$$

where $\pi_i(\mathcal{C}_i)$ is a sub-policy of agent i for the predecessor coalition $\mathcal{C}_i$.

Suppose for the sake of contradiction that we have $\max_{\pi_{\mathcal{C}}} \tilde{x}(\mathbf{s}|\mathcal{C}) < \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) - \epsilon(\mathcal{C})$ for some $\mathbf{s}\in\mathcal{S}$ and some coalition $\mathcal{C} = \{j_1, j_2, ..., j_{|\mathcal{C}|}\}\subseteq\mathcal{N}$, where $j_n\in\mathcal{C}$, $n\in\{1,2,...,|\mathcal{C}|\}$ and

$$\epsilon(\mathcal{C}) = (1-\frac{|\mathcal{C}|!}{|\mathcal{N}|!})\cdot \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}).$$

First, we define $\mathbb{P}(\mathcal{N}\backslash\{i\})$ as the power set of $\mathcal{N}\backslash\{i\}$ and $\mathbb{P}(\mathcal{C}\backslash\{i\})$ as the power set of $\mathcal{C}\backslash\{i\}$. For the convenience of proof, we rewrite the Eq.50 to the form such that

$$\max_{\pi_i} v_i^{\phi}(\mathbf{s}) = \underbrace{\sum_{\mathcal{C}_i\subseteq\mathcal{C}\backslash\{i\}} w_{\mathcal{C}_i}\cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i)}_{\text{The part correlated to } \mathcal{C}.} + \underbrace{\sum_{\mathcal{C}_i\in\mathbb{P}(\mathcal{N}\backslash\{i\})\backslash\mathbb{P}(\mathcal{C}\backslash\{i\})} w_{\mathcal{C}_i}\cdot \max_{\pi_i(\mathcal{C}_i)} \Phi_i(\mathbf{s}|\mathcal{C}_i)}_{\text{The part uncorrelated to } \mathcal{C}.}, \tag{51}$$

where $w_{\mathcal{C}_i}$ is the fraction of occurrence frequency (e.g. $\frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}$) for their correlated coalitional marginal contributions.

For each sequence $m = \langle j_1, j_2, ..., j_{|\mathcal{C}|}\rangle$ generated from $\mathcal{C}$, thanks to the result from Theorem 3 we have an equation such that

$$\sum_{i\in\mathcal{C}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) \geq \max_{\pi_{\mathcal{C}^m}} v^{\pi_{\mathcal{C}^m}}(\mathbf{s}), \ \forall \mathbf{s} \in \mathcal{S}, \tag{52}$$

where $\Phi_i(\mathbf{s}|\mathcal{C}_i^m)$ denotes the coalitional marginal contribution of agent i belonging to the coalition $\mathcal{C}$ for the predecessor coalition $\mathcal{C}_i^m$ of the agent in this sequence m and $\pi_{\mathcal{C}^m} = \times_{i\in\mathcal{C}}\pi_i(\mathcal{C}_i^m)$ denotes the joint policy of agents for sequence m.

If we add the inequalities of all sequences, we can get the equation as follows:

$$\sum_{m\in\Pi(\mathcal{C})}\sum_{i\in\mathcal{C}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) \geq \sum_{m\in\Pi(\mathcal{C})} \max_{\pi_{\mathcal{C}^m}} v^{\pi_{\mathcal{C}^m}}(\mathbf{s}), \ \forall \mathbf{s} \in \mathcal{S}, \tag{53}$$

where $\Pi(\mathcal{C})$ indicates the set of all permutations generated from the coalition $\mathcal{C}$.

Dividing $|\mathcal{N}|!$ on both sides of Eq.53, we can get the equation such that

$$\frac{1}{|\mathcal{N}|!}\sum_{m\in\Pi(\mathcal{C})}\sum_{i\in\mathcal{C}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) \geq \frac{1}{|\mathcal{N}|!}\sum_{m\in\Pi(\mathcal{C})} \max_{\pi_{\mathcal{C}^m}} v^{\pi_{\mathcal{C}^m}}(\mathbf{s}), \ \forall \mathbf{s} \in \mathcal{S}. \tag{54}$$

To avoid confusion, we firstly process the left hand side of Eq.54 as follows:

$$\begin{aligned}
\frac{1}{|\mathcal{N}|!}\sum_{m\in\Pi(\mathcal{C})}\sum_{i\in\mathcal{C}} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) &= \sum_{i\in\mathcal{C}}\frac{1}{|\mathcal{N}|!}\sum_{m\in\Pi(\mathcal{C})} \max_{\pi_i(\mathcal{C}_i^m)} \Phi_i(\mathbf{s}|\mathcal{C}_i^m) \\
&\quad \text{(The identical } \mathcal{C}_i^m \text{ in different sequences is written as } \mathcal{C}_i \\
&\quad \ \text{and we can rearrange the equation as follows.)} \\
&= \sum_{i\in\mathcal{C}}\frac{1}{|\mathcal{N}|!}\sum_{\mathcal{C}_i\subseteq\mathcal{C}\backslash\{i\}} |\mathcal{C}_i|!(|\mathcal{C}|-|\mathcal{C}_i|-1)!\cdot\max_{\pi_i(\mathcal{C}_i)}\Phi_i(\mathbf{s}|\mathcal{C}_i) \\
&= \sum_{i\in\mathcal{C}}\sum_{\mathcal{C}_i\subseteq\mathcal{C}\backslash\{i\}} \frac{|\mathcal{C}_i|!(|\mathcal{C}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}\cdot\max_{\pi_i(\mathcal{C}_i)}\Phi_i(\mathbf{s}|\mathcal{C}_i) \\
&\text{(We use } w_{\mathcal{C}_i} \text{ here to represent each } \frac{|\mathcal{C}_i|!(|\mathcal{C}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}.\text{)} \\
&= \sum_{i\in\mathcal{C}}\sum_{\mathcal{C}_i\subseteq\mathcal{C}\backslash\{i\}} w_{\mathcal{C}_i}\cdot\max_{\pi_i(\mathcal{C}_i)}\Phi_i(\mathbf{s}|\mathcal{C}_i).
\end{aligned} \tag{55}$$

Next, we process the right hand side of Eq.54 as follows. By Assumption 1, we can get that each $v^{\pi_{\mathcal{C}^m}}$ is identical. By Assumption 2, we can use a generalised joint policy $\pi_{\mathcal{C}} = \times_{i\in\mathcal{C}}\pi_i$ to include $\biguplus_{m\in\Pi(\mathcal{C})}\pi_{\mathcal{C}^m}$, since $\pi_i = \biguplus_{\mathcal{C}_i\subseteq\mathcal{N}\backslash\{i\}}\pi_i(\mathcal{C}_i) \supseteq \biguplus_{m\in\Pi(\mathcal{C})}\pi_i(\mathcal{C}_i^m)$ and $\biguplus_{m\in\Pi(\mathcal{C})}\pi_{\mathcal{C}^m} = \biguplus_{m\in\Pi(\mathcal{C})}\times_{i\in\mathcal{C}}\pi_i(\mathcal{C}_i^m) = \times_{i\in\mathcal{C}}\biguplus_{m\in\Pi(\mathcal{C})}\pi_i(\mathcal{C}_i^m)$. Therefore, we can get the following results such that

$$\begin{aligned}
\frac{1}{|\mathcal{N}|!}\sum_{m\in\Pi(\mathcal{C})}\max_{\pi_{\mathcal{C}^m}} v^{\pi_{\mathcal{C}^m}}(\mathbf{s}) &= \frac{1}{|\mathcal{N}|!}\sum_{m\in\Pi(\mathcal{C})}\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) \\
&= \frac{|\mathcal{C}|!}{|\mathcal{N}|!}\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}).
\end{aligned} \tag{56}$$

Combining the results from Eq.55 and 56, we can get the equation as follows:

$$\sum_{i\in\mathcal{C}}\sum_{\mathcal{C}_i\subseteq\mathcal{C}\backslash\{i\}} w_{\mathcal{C}_i}\cdot\max_{\pi_i(\mathcal{C}_i)}\Phi_i(\mathbf{s}|\mathcal{C}_i) \geq \frac{|\mathcal{C}|!}{|\mathcal{N}|!}\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}). \tag{57}$$

By the result from Proposition 4, it is trivial that $\max_{\pi_i(\mathcal{C}_i)}\Phi_i(\mathbf{s}|\mathcal{C}_i) \geq 0$ always holds. For this reason,

$$\Delta = \sum_{\mathcal{C}_i\in\mathbb{P}(\mathcal{N}\backslash\{i\})\backslash\mathbb{P}(\mathcal{C}\backslash\{i\})} w_{\mathcal{C}_i}\cdot\max_{\pi_i(\mathcal{C}_i)}\Phi_i(\mathbf{s}|\mathcal{C}_i) \geq 0$$

also always holds. Adding Δ to the left hand side of Eq.57, by Assumption 2 we can get the equations such that

$$\begin{aligned}
\sum_{i\in\mathcal{C}}\sum_{\mathcal{C}_i\subseteq\mathcal{C}\backslash\{i\}} w_{\mathcal{C}_i}\cdot\max_{\pi_i(\mathcal{C}_i)}\Phi_i(\mathbf{s}|\mathcal{C}_i)+\Delta &\geq \frac{|\mathcal{C}|!}{|\mathcal{N}|!}\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) \\
&\Downarrow \\
\sum_{i\in\mathcal{C}}\max_{\pi_i} v_i^{\phi}(\mathbf{s}) = \max_{\pi_{\mathcal{C}}}\tilde{x}(\mathbf{s}|\mathcal{C}) &\geq \frac{|\mathcal{C}|!}{|\mathcal{N}|!}\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) \\
&= \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) - \frac{|\mathcal{N}|!-|\mathcal{C}|!}{|\mathcal{N}|!}\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) \\
&= \max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}) - \epsilon(\mathcal{C}). \qquad (58)
\end{aligned}$$

It is apparent that the result from Eq.58 contradicts the suppose and we show that the existence of the ϵ-core, where

$$\epsilon(\mathcal{C}) = (1-\frac{|\mathcal{C}|!}{|\mathcal{N}|!})\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s}). \qquad (59)$$

Next, to find the least core we continue to analyse the $\epsilon(\mathcal{C})$ to find a ϵ^* that can make Eq.58 hold for any coalition, thereby finding the ϵ^*-core.

Since $(1-\frac{|\varnothing|!}{|\mathcal{N}|!})\cdot v^{\pi_{\varnothing}}(\mathbf{s})=0$ and $(1-\frac{|\mathcal{N}|!}{|\mathcal{N}|!})\cdot\max_{\pi} v^{\pi}(\mathbf{s})=0$ as well as the fact that $v^{\pi_{\mathcal{C}}}(\mathbf{s})\geq 0$ for any $\mathcal{C}$ and $\mathbf{s}\in\mathcal{S}$, it is obvious that there exists a supremum between the coalition $\varnothing$ and $\mathcal{N}$.

For any $\mathcal{C}\subseteq\mathcal{N}$, we can get the following upper bound for $\epsilon(\mathcal{C})$ such that

$$\epsilon(\mathcal{C}) \leq \sup_{\mathcal{C}\in\mathbb{P}(\mathcal{N})\backslash\{\mathcal{N},\varnothing\}}\Big\{(1-\frac{|\mathcal{C}|!}{|\mathcal{N}|!})\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s})\Big\}. \qquad (60)$$

Henceforth, we can get that

$$\begin{aligned}
&\inf\Big\{\epsilon \mid \max_{\pi_{\mathcal{C}}}\tilde{x}(\mathbf{s}|\mathcal{C})\geq\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s})-\epsilon,\ \forall\mathbf{s}\in\mathcal{S},\forall\mathcal{C}\subseteq\mathcal{N}\Big\} \\
&= \sup_{\mathcal{C}\in\mathbb{P}(\mathcal{N})\backslash\{\mathcal{N},\varnothing\}}\Big\{(1-\frac{|\mathcal{C}|!}{|\mathcal{N}|!})\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s})\Big\}.
\end{aligned} \qquad (61)$$

Therefore, we prove that the generalised Shapley value is a solution in the ϵ^*-core, where

$$\epsilon^* = \sup_{\mathcal{C}\in\mathbb{P}(\mathcal{N})\backslash\{\mathcal{N},\varnothing\}}\Big\{(1-\frac{|\mathcal{C}|!}{|\mathcal{N}|!})\cdot\max_{\pi_{\mathcal{C}}} v^{\pi_{\mathcal{C}}}(\mathbf{s})\Big\}. \qquad (62)$$

□

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

Derivation of Shapley-Q Optimality Equation. First, according to Bellman's principle of optimality (Bellman, 1952; Sutton & Barto, 2018) we can write out Bellman optimality equation for the global Q-value such that

$$Q_*^{\pi}(\mathbf{s},\mathbf{a}) = \sum_{\mathbf{s}'} p(\mathbf{s}'|\mathbf{s},\mathbf{a})\big[R+\gamma\max_{\mathbf{a}} Q_*^{\pi}(\mathbf{s}',\mathbf{a})\big]. \qquad (63)$$

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 get that

$$\max_{\mathbf{a}} Q_*^{\pi}(\mathbf{s}',\mathbf{a}) = \sum_{i\in\mathcal{N}}\max_{a_i} Q_i^{\phi^*}(\mathbf{s}',a_i). \qquad (64)$$

As per the theoretical results in Theorem 1, the grand coalition with the generalised Shapley (Q-)value as the value assignment scheme always stays in the ϵ^*-core. For any subsequence of decisions (subgame), almost no agents would have large incentives to deviate from the grand coalition or change the value assignment scheme, since the current strategy (i.e., coalition structure and value assignment scheme) can lead to the maximal individual value assignment and the maximal social welfare. Standing by the side of solution concept, any subsequence of a sequence of decisions in the ϵ-core is still in the ϵ-core. Since the solution in the ϵ-core can lead to the optimal social welfare (i.e., the said optimal global Q-value), this is consistent with the fact that the optimal global Q-value over a time step should be built upon the optimal global Q-value over any subsequent time steps, as per the principle of optimality (Bellman, 1952). Literally, we can simply think of that all agents at the beginning of a sequence of decisions are rational and remain the grand coalition and the generalised Shapley value as the value assignment until the end of game. The above discussion implies the regularity of Eq.64.

By definition in Section 2, $Q_*^{\pi}(\mathbf{s}, \mathbf{a}) > 0$ holds. Since Proposition 4 and Eq.37, we can get that $Q_i^{\phi^*}(\mathbf{s}, a_i) \geq 0$. By using a extremely small number (i.e. 10^{-7}) to approximate $Q_i^{\phi^*}(\mathbf{s}, a_i) = 0$, we can get that $Q_i^{\phi^*}(\mathbf{s}, a_i) \geq 10^{-7}$. It is a fact that for all $\mathrm{s} \in \mathcal{S}$ and $a_i \in \mathcal{A}_i$, there exists a $w_i(\mathbf{s}, a_i) > 0$ such that

$$Q_i^{\phi^*}(\mathbf{s}, a_i) = w_i(\mathbf{s}, a_i)\ Q_*^{\pi}(\mathbf{s}, \mathbf{a}). \tag{65}$$

If we denote $\mathbf{w}(\mathbf{s}, \mathbf{a}) = [w_i(\mathbf{s}, a_i)]^\top$ and $\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = [Q_i^{\phi^*}(\mathbf{s}, a_i)]^\top$, given Eq.65 we can write that

$$\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = \mathbf{I}\ \mathbf{w}(\mathbf{s}, \mathbf{a})\ Q_*^{\pi}(\mathbf{s}, \mathbf{a}), \tag{66}$$

where $\mathbf{I}$ is an identity matrix. Combined with Eq.64 and 66, we can rewrite Eq.63 to the equation as follows:

$$\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a}) = \mathbf{I}\ \mathbf{w}(\mathbf{s}, \mathbf{a}) \sum_{\mathbf{s}'} p(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \big[R \ + \ \gamma \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}', a_i) \ \big]. \tag{67}$$

From Eq.65, we know that $w_i(\mathbf{s}, a_i) > 0$. Therefore, we can rewrite Eq.65 to the following equation such that

$$w_i(\mathbf{s}, a_i)^{-1}\ Q_i^{\phi^*}(\mathbf{s}, a_i) = Q_*^{\pi}(\mathbf{s}, \mathbf{a}). \tag{68}$$

If we sum up Eq.68 for all agents, we can obtain that

$$\sum_{i \in \mathcal{N}} w_i(\mathbf{s}, a_i)^{-1}\ Q_i^{\phi^*}(\mathbf{s}, a_i) = |\mathcal{N}|\ Q_*^{\pi}(\mathbf{s}, \mathbf{a}). \tag{69}$$

Therefore, we can get the following equation such that

$$\sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|\ w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi^*}(\mathbf{s}, a_i) = \ Q_*^{\pi}(\mathbf{s}, \mathbf{a}). \tag{70}$$

Substituting Eq.70 for $Q_*^{\pi}(\mathbf{s}, \mathbf{a})$ in Eq.64, we can get the following equation such that

$$\max_{\mathbf{a}} \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|\ w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi^*}(\mathbf{s}, a_i) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i). \tag{71}$$

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

$$\sum_{i \in \mathcal{N}} \max_{a_i} \frac{1}{|\mathcal{N}|\ w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi^*}(\mathbf{s}, a_i) = \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi^*}(\mathbf{s}, a_i). \tag{72}$$

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

Lemma 2 (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 3. *For a set of real matrices $\mathcal{M}$, given an arbitrary matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$, $||\mathbf{A}||_1 = \max_{1 \leq j \leq n} \sum_{1 \leq i \leq m} |a_{ij}|$ is a sub-multiplicative norm.*

Proof. The complete proof is as follows.

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

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

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 4. *For all* $\mathbf{s} \in \mathcal{S}$ *and* $\mathbf{a} \in \mathcal{A}$*, Shapley-Q 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

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

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

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

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

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

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

$$
\begin{aligned}
& ||\Upsilon \mathbf{Q}_1^{\phi} - \Upsilon \mathbf{Q}_2^{\phi}||_1 \\
&= \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^{\top} \Big| \mathbf{I}\,\mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \big[R(\mathbf{s},\mathbf{a}) + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) \big] \\
&- \mathbf{I}\,\mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \big[R(\mathbf{s},\mathbf{a}) + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \big] \Big| \\
&= \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^{\top} \Big| \mathbf{I}\,\mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \big[R(\mathbf{s},\mathbf{a}) + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) \\
&- R(\mathbf{s},\mathbf{a}) - \gamma \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \big] \Big| \\
&= \gamma \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^{\top} \Big| \mathbf{I}\,\mathbf{w}(\mathbf{s},\mathbf{a}) \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \big[\sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \big] \Big| \\
&\leq \gamma \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^{\top} \Big| \mathbf{I}\,\mathbf{w}(\mathbf{s},\mathbf{a})\ \Big| \max_{\mathbf{s},\mathbf{a}} \Big| \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \big[\sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \big] \Big|
\end{aligned}
$$

(If we write $\delta = \gamma \max_{\mathbf{s},\mathbf{a}} \mathbf{1}^{\top} \big| \mathbf{I}\,\mathbf{w}(\mathbf{s},\mathbf{a})\ \big|$, we can have the following equation.)

$$
\begin{aligned}
&= \delta \max_{\mathbf{s},\mathbf{a}} \Big| \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \big[\sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \big] \Big| \\
&\leq \delta \max_{\mathbf{s},\mathbf{a}} \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \Big| \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \Big| \\
&= \delta \Big| \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \sum_{i\in\mathcal{N}} \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \Big| \\
&= \delta \Big| \sum_{i\in\mathcal{N}} \big[\max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \big] \Big|
\end{aligned}
$$

(By triangle inequality, we can obtain the following inequality.)

$$
\begin{aligned}
&\leq \delta \sum_{i\in\mathcal{N}} \Big| \max_{a_i} \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \max_{a_i} \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \Big| \\
&\leq \delta \sum_{i\in\mathcal{N}} \max_{a_i} \Big| \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \Big|
\end{aligned}
$$

(Since $\mathbf{a} = \times_{i\in\mathcal{N}} a_i$, we have the following equation.)

$$
\begin{aligned}
&= \delta \max_{\mathbf{a}} \sum_{i\in\mathcal{N}} \Big| \big(Q_i^{\phi}\big)_1(\mathbf{s}',a_i) - \big(Q_i^{\phi}\big)_2(\mathbf{s}',a_i) \Big| \\
&\leq \delta \max_{\mathbf{z},\mathbf{a}} \sum_{i\in\mathcal{N}} \Big| \big(Q_i^{\phi}\big)_1(\mathbf{z},a_i) - \big(Q_i^{\phi}\big)_2(\mathbf{z},a_i) \Big| \\
&= \delta ||\mathbf{Q}_1^{\phi} - \mathbf{Q}_2^{\phi}||_1 .
\end{aligned}
$$

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

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

Therefore, we show that Shapley-Q 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. □

Remark 5. *It is not difficult to extend the result from Theorem 4 to other measure spaces (e.g., Lebesgue measure space) for the infinite $\mathcal{S}$ and $\mathcal{A}$ instead of the counting measure for the finite $\mathcal{S}$ and $\mathcal{A}$ that we used here. Since the proof sketch is similar, we ignore the proof here.*

Corollary 1. *According to Banach fixed-point theorem (Banach, 1922), Shapley-Q operator admits a unique fixed point. Moreover, starting by an arbitrary start point, the sequence recursively generated by Shapley-Q 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-Q operator Υ is shown as a contraction mapping in Lemma 4, by Banach fixed-point theorem (Banach, 1922) we can directly conclude that Shapley-Q operator Υ admits a unique fixed point. Furthermore, starting by an arbitrary start point, the sequence recursively generated by Shapley-Q operator Υ can finally converge to that fixed point. □

Theorem 2. *Shapley-Q operator can converge to the optimal Shapley Q-values and therefore the optimal joint deterministic policy is achieved when* $\max_{\mathbf{s}}\big\{\sum_{i\in\mathcal{N}}\max_{a_i} w_i(\mathbf{s},a_i)\big\} < \frac{1}{\gamma}$.

Proof. By Corollary 1, we get that Shapley-Q operator admits a unique fixed point. Since Eq.7 is obviously a fixed point for Shapley-Q operator, it is not difficult to conclude that the optimal Shapley Q-value can be converged to and the optimal joint deterministic policy is achieved. □

Stochastic Approximation of Shapley-Q Operator. We now derive the stochastic approximation of Shapley-Q operator, i.e. a form of Q-learning derived from Shapley-Q operator. By sampling from $p(s'|s,a)$, the Q-learning algorithm can be expressed as follows:

$$\mathbf{Q}_{t+1}^{\phi}(\mathbf{s},\mathbf{a}) \leftarrow \mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a}) + \alpha_t(\mathbf{s},\mathbf{a})\Big[\mathbf{Iw}(\mathbf{s},\mathbf{a})\Big(R_t + \gamma\sum_{i\in\mathcal{N}}\max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\Big) - \mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a})\Big]. \quad (73)$$

Lemma 5 (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***$[F_t(x)|\mathcal{F}_t] \le C(1 + ||\Delta_t||_W^2)$, *for* $C > 0$.

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

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

converges w.p.1 to the optimal generalised Shapley Q-values if

$$\sum_t \alpha_t(\mathbf{s},\mathbf{a}) = \infty \qquad \sum_t \alpha_t^2(\mathbf{s},\mathbf{a}) \le \infty \quad (74)$$

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

Proof. The proof follows the sketch of proving the convergence of Q-learning given by Melo (2001). First, we rewrite Eq.73 to

$$\mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a}) = (1-\alpha_t(\mathbf{s},\mathbf{a}))\mathbf{Q}_t^{\phi}(\mathbf{s},\mathbf{a}) + \alpha_t(\mathbf{s},\mathbf{a})\Big[\mathbf{Iw}(\mathbf{s},\mathbf{a})\Big(R_t + \gamma\sum_{i\in\mathcal{N}}\max_{a_i}(Q_i^{\phi})_t(\mathbf{s}',a_i)\Big)\Big].$$

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

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

we can transform Eq.73 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

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

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

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

By the results from Theorem 4, we can get that

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

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

Next, we get that

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

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

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

for some constant C.

Finally, by Lemma 5 it is easy to see that Δ_t converges to 0 w.p.1, i.e., $\mathbf{Q}_t^{\phi}(\mathbf{s}, \mathbf{a})$ converges to $\mathbf{Q}^{\phi^*}(\mathbf{s}, \mathbf{a})$ w.p.1, given the condition in Eq.74. □

Derivation of Shapley Q-Learning. By stochastic approximation, i.e. sampling $\mathbf{s}'$ from $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, Shapley-Q operator can be expressed as follows:

$$\mathbf{Q}^{\phi}(\mathbf{s}, \mathbf{a}) = \mathbf{Iw}(\mathbf{s}, \mathbf{a})\big(R + \gamma \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i)\big), \tag{75}$$

where $\mathbf{I}$ is an identity matrix; $\mathbf{w}(\mathbf{s}, \mathbf{a}) = [w_i(\mathbf{s}, a_i)]^\top \in \mathbb{R}_+^{|\mathcal{N}|}$; and $\mathbf{Q}^{\phi}(\mathbf{s}, \mathbf{a}) = [Q_i^{\phi}(s, a_i)]^\top \in \mathbb{R}_+^{|\mathcal{N}|}$. Eq.75 can be equivalently represented as

$$\big(\mathbf{Iw}(\mathbf{s}, \mathbf{a})\big)^{-1}\mathbf{Q}^{\phi}(\mathbf{s}, \mathbf{a}) = \mathbf{1}\big(R + \gamma \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i)\big), \tag{76}$$

where $\mathbf{1}$ is a vector of ones. Next, we multiply $\mathbf{1}^\top$ on both sides and obtain the following equation such that

$$\sum_{i \in \mathcal{N}} \frac{1}{w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi}(s, a_i) = |\mathcal{N}| \ \big(R + \gamma \sum_{i \in \mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i)\big). \tag{77}$$

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

$$\sum_{i\in\mathcal{N}} \frac{1}{|\mathcal{N}| w_i(\mathbf{s}, a_i)} \cdot Q_i^{\phi}(s, a_i) = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(s, a_i). \tag{78}$$

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

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \end{cases} \tag{79}$$

where $\alpha_i(\mathbf{s}, a_i) = \frac{1}{|\mathcal{N}|\ w_i(\mathbf{s}, a_i)}$ for $a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i)$.

By substituting Eq79 to Eq.78, we can get the following equation such that

$$\sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\ Q_i^{\phi}(\mathbf{s}, a_i) = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i). \tag{80}$$

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

$$\Delta(\mathbf{s}, \mathbf{a}, \mathbf{s}') = R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} Q_i^{\phi}(\mathbf{s}', a_i) - \sum_{i\in\mathcal{N}} \delta_i(\mathbf{s}, a_i)\ Q_i^{\phi}(\mathbf{s}, a_i). \tag{81}$$

The TD error for SHAQ is necessary for the TD error for Eq.73 (i.e. the stochastic learning process that we proved to converge to the optimal generalised Shapley Q-value in Theorem 4). For this reason, the condition $\max_{\mathbf{s}} \big\{ \sum_{i\in\mathcal{N}} \max_{a_i} w_i(\mathbf{s}, a_i) \big\} < \frac{1}{\gamma}$ needs to be satisfied so that the convergence to the optimality is possible to hold.

B.5 Mathematical Derivation for Implementation of Shapley Q-Learning

Proposition 3. *Suppose any coalitional 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*

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

we have

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

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

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 generalised 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:

$$\delta_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i), \\ \alpha_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} Q_i^{\phi}(\mathbf{s}, a_i). \end{cases} \tag{82}$$

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

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

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. □

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 $\mathcal{C}_i^k \sim p(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ 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.11 shows, it is not difficult to get that $\alpha_i(\mathbf{s}, a_i) = K^{-1}\ \hat{\alpha}_i(\mathbf{s}, a_i)$. Since $K \in (0, 1)$, we get that $\alpha_i(\mathbf{s}, a_i) > 1$.

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

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

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

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

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

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

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

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

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

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

We get that our implementation of $\hat{\alpha}_i(\mathbf{s}, a_i)$ satisfies the condition in Theorem 2.

C 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.

Implementation of Sampling from $p(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ (Line 4 in Algorithm 2). As introduced before, the analytic form of $p(\mathcal{C}_i|\mathcal{N}\backslash\{i\})$ is $\frac{|\mathcal{C}_i|!(|\mathcal{N}|-|\mathcal{C}_i|-1)!}{|\mathcal{N}|!}$ that is actually the occurrence frequency of

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 $\langle \mathbf{s}^t, (o_i^t)_{i=1:N}, (a_i^t)_{i=1:N}, R^t, \mathbf{s}^{t+1}, (o_i^{t+1})_{i=1:N} \rangle$ 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 $\langle \mathbf{s}^k, (o_i^k)_{i=1:N}, (a_i^k)_{i=1:N}, R^k, \mathbf{s}^{k+1}, (o_i^{k+1})_{i=1:N} \rangle$
            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:
                $\hat{\delta}_i(\mathbf{s}^k, a_i^k; \lambda) = \begin{cases} 1 & a_i^k = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}^k, a_i; \theta) \\ \hat{\alpha}_i(\mathbf{s}^k, a_i^k; \lambda) & a_i^k \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}^k, a_i; \theta) \ \text{(via Algorithm 2)} \end{cases}$
            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 $\theta$ and $\lambda$ through the above loss
    Periodically update $\theta^-$ by copying $\theta$
until $\hat{Q}_i(\tau_i, a_i; \theta)$ converges
```

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

```
Input: s, $\big(\hat{Q}_i(\tau_i, a_i; \theta)\big)_{i=1:N}$, $M$
Output: $\big(\hat{\alpha}_i(\mathbf{s}, a_i)\big)_{i=1:N}$
for each agent $i$ do
    Sample $M$ preceding coalitions $\mathcal{C}_i^k \sim p(\mathcal{C}_i | \mathcal{N} \backslash \{i\})$
    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 $\alpha_i(\mathbf{s}, a_i) = \frac{1}{M} \sum_{k=1}^{M} F_{\mathbf{s}}\Big(\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}), \ \hat{Q}_i(\tau_i, a_i)\Big) + 1$
end for
```

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 coalitional marginal contributions and therefore the approximate generalised Shapley value is obtained.

D Experimental Setups

D.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.

D.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.

D.3 Predator-Prey for Modelling Relative Overgeneralisation

We give the experimental setups of Predator-Prey (Böhmer et al., 2020) in Table 2.

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

D.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 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

E Extra Experimental Results

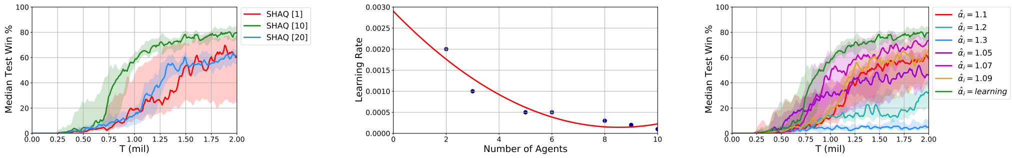
E.1 Ablation Study for SHAQ

Sample Size M for Approximating $\hat{\alpha}(\mathrm{s}, a_i)$. To study the impact of sample size M on the performance of SHAQ, we conduct an ablation study as Figure 4a 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 for Selecting the Learning Rate for $\hat{\alpha}_i(s, a_i)$. To provide an empirical law on selecting the learning rate for $\hat{\alpha}_i(s, a_i)$, we statistically fit a curve of the learning rate w.r.t. the

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



(a) Comparison among different 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.

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

Figure 4: The figures of 3 ablation studies for SHAQ on SMAC.

number of controllable agents by the experimental results on SMAC that is shown in Figure 4b. It is seen that the learning rate for $\hat{\alpha}_i(s, a_i)$ is generally negatively related to the number of agents. In other words, as the number of agents grows, the learning rate for $\hat{\alpha}_i(s, a_i)$ is recommended to be smaller. For example, if the number of agents is more than 10, the learning rate for $\hat{\alpha}_i(s, a_i)$ is recommended to be 0.0001 as the guidance from Figure 4b.

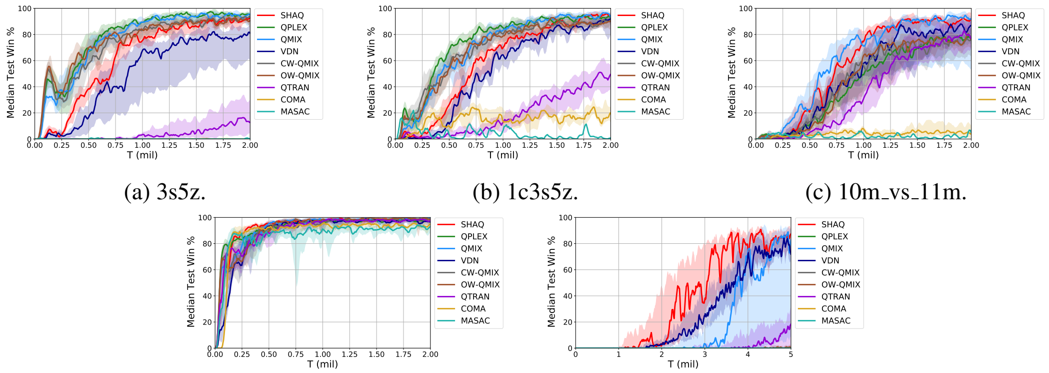
The Necessity of Learning $\hat{\alpha}_i(\mathbf{s}, a_i)$. Some readers may be concerned about the necessity of learning $\hat{\alpha}_i(\mathbf{s}, a_i)$. To answer this question, we study the necessity of learning $\hat{\alpha}_i(\mathbf{s}, a_i)$ on 5m_vs_6m. Since the learned $\hat{\alpha}_i(\mathbf{s}, a_i)$ finally converges to 1.1029, we grid search the fixed values of $\hat{\alpha}_i(\mathbf{s}, a_i)$ around this number. As Figure 4c 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.

E.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 5 shows. 8m, 3s5z, 1c3s5z and 10m_vs_11m are an easy maps and corridor is a super-hard map. The strategy of epsilon annealing is consistent with the previous experiments for SMAC. It is obvious that SHAQ also performs generally well on these 5 maps.

E.3 Extra Experimental Results on W-QMIX with $\alpha = \mathbf{0.1}$

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 6 that the performances of W-QMIX are not comparatively identical for each choice of α. As a result, W-QMIX suffers from the separate tuning of α for each scenario. Unfortunately, Rashid et al. (2020) did not provide an empirical law for selecting α, while SHAQ possesses an empirical law to select $\hat{\alpha}_i(\mathbf{s}, a_i)$ as Figure 4c shows.



(a) 3s5z.

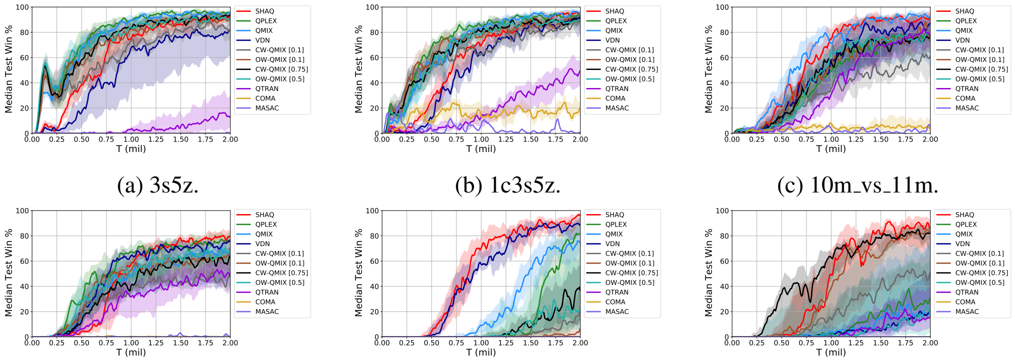
(b) 1c3s5z.

(c) 10m_vs_11m.

(d) 8m.

(e) Corridor.

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



(a) 3s5z.

(b) 1c3s5z.

(c) 10m_vs_11m.

(d) 5m_vs_6m.

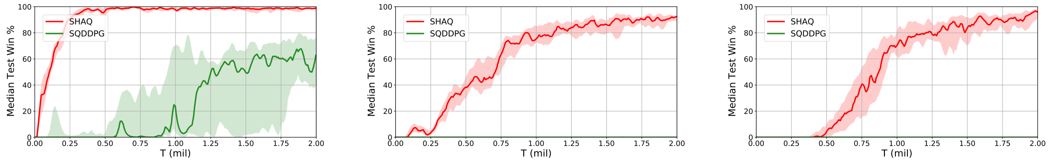
(e) 3s_vs_5z.

(f) 2c_vs_64zg.

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

E.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. As Figure 7 shows, the performance of SHAQ surpasses that of SQDDPG on all 3 maps, while SQDDPG can only learn on the simplest map 3m. The most possible reason for the failure of SQDDPG to complicated tasks is its sample complexity inefficiency for permutations of agents as discussed in Section 5 that leads to the difficulty in learning. Apparently, the implementation of coalition invariance of SHAQ mitigates this weakness so that it is able to solve more challenging tasks.

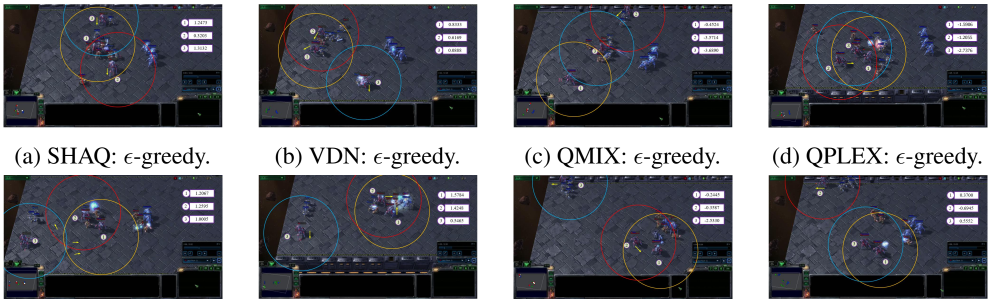


(a) 3m.

(b) 3s5z.

(c) 3s_vs_5z.

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



(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 8: 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.

E.5 More Visualisation

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 8. First, we take a look into the optimal decisions (from greedy decision). 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 sub-optimal decisions. Therefore, the parametric effects of the mistakes conducted on sub-optimal 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 strategies, the agent who is responsible for kiting [4] (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 mixture of optimal and sub-optimal decisions (from ϵ-greedy decision). 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 regarded as a dummy agent and therefore obtains the credit near 0. This is consistent again 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.

E.6 Extra Experimental Results for Predator-Prey

In Figure 9, we show the results for W-QMIX with p=-0.5,-1,-2 and the annealing steps as 50k to support our claims in the experimental analysis that the poor performance of W-QMIX on Predator-Prey is due to its poor robustness to the increased explorations for this environment. We also show the performance of SQDDPG with p=-2 and the epsilon annealing steps as 1 mil.

F Extra Related Works

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 methods above are in static scenarios that just directly used the original Shapley value theory for application. However, our work extends the Shapley value theory to Markov dynamics with action space. The

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

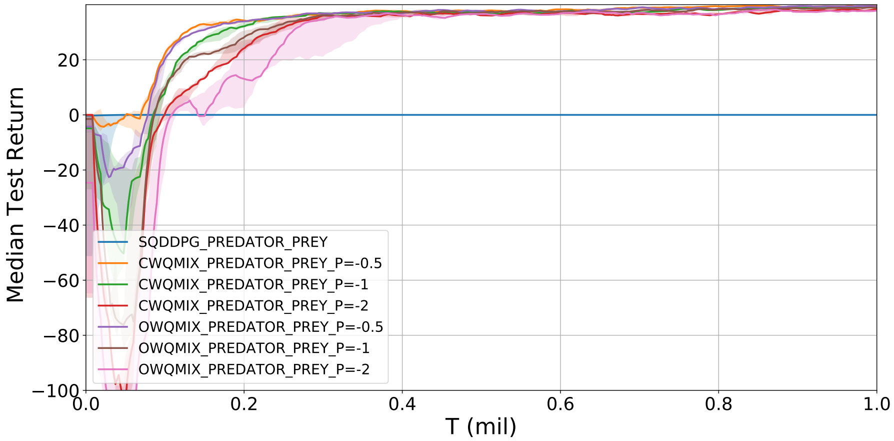


Figure 9: Median test return for Predator-Prey evaluated with SQDDPG and W-QMIX (inlcuding CW-QMIX and OW-QMIX). For SQDDPG, p=-2 and the epsilon annealing steps are 1 mil. For W-QMIX, we evaluate the performances on p=-0.5,-1,-2 and the epsilon annealing steps are 50k.

theoretical framework of Shapley value for MCG with the grand coalition proposed in this paper is easy to be applied for designing MARL algorithms with interpretability.

Learning Paradigm for MARL. Considering the algorithmic frameworks among MARL algorithms with CTDE, it can be classified to two categories such as multi-agent Q-learning algorithms (Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019; Wang et al., 2020b; 2021) and multi-agent actor-critic algorithms (Lowe et al., 2017; Foerster et al., 2018; Iqbal & Sha, 2019; Wang et al., 2020c; Mahajan et al., 2021). In this paper, we propose Shapley Q-learning (SHAQ) that belongs to the category of multi-agent Q-learning algorithms.