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

Anonymous Author(s)
Affiliation
Address
email

Abstract

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

1 Introduction

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

By the view of non-cooperative game theory, global reward game are equivalent to Markov game [12] with global reward (a.k.a. team reward). Its aim is to learn a stationary joint policy to reach a Markov equilibrium so that no agent tends to unilaterally change its policy to maximize cumulative global rewards. Standing by this view, learning with value factorisation cannot be directly explained [13]. In this paper, to clearly interpret the value factorisation, we take the perspective of cooperative game theory [14], wherein agents are partitioned into coalitions and a payoff distribution scheme is found to distribute optimal values to coalitions. The corresponding solution is called Markov core, whereby no agent has an incentive to deviate. When all agents are partitioned into one coalition (called grand coalition), the payoff distribution scheme naturally plays the role of value factorisation for an optimal global value.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

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

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

The proposed method, SHAQ, is evaluated on two global reward games such as Predator-Prey [17] and multi-agent StarCraft benchmark tasks [18]. In the experiments, SHAQ shows not only generally good performances on solving all tasks but also the interpretability that is deficient in the state-of-the-art baselines. Specifically, the learned Markov Shapley Q-value reflects the contribution of each agent and identifies dummy agents. Besides, the optimal Shapley Q-value of each agent is demonstrated equal (i.e., a sort of fairness). All of these support our theoretical analyses.

2 Markov Convex Game

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

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

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

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

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

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

3 Markov Shapley Value

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

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

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

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

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

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

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

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

Definition 2. *Markov Shapley value is represented as*

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

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

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

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

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

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

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

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

4 Shapley Q-Learning

4.1 Definition and Formulation

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

C.1. Efficiency of MSV obtained from Proposition 2;

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

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

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

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

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

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

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

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

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

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

where

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

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

4.2 Validity and Interpretability

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

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

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

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

4.3 Implementations

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

Compatible with the decentralised execution, we use only one parametric function $\hat{Q}_i(\tau_i, a_i)$ to directly approximate $Q_i^{\phi}(\tau_i, a_i)$. With specific conditions (see Proposition 8 in Appendix E.6) the information of coalition formation can be equivalently transferred to $\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})$. As a result, $\delta_i(\mathbf{s}, a_i)$ is equivalent to the form as follows:

$$\hat{\delta}_i(\mathbf{s}, a_i) = \begin{cases} 1 & a_i = \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \\ \hat{\alpha}_i(\mathbf{s}, a_i) & a_i \neq \arg\max_{a_i} \hat{Q}_i(\mathbf{s}, a_i), \end{cases} \tag{11}$$

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

$$\hat{\alpha}_i(\mathbf{s}, a_i) = \frac{1}{M} \sum_{k=1}^{M} F_{\mathbf{s}}\Big(\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}),\ \hat{Q}_i(\tau_i, a_i)\Big) + 1, \tag{12}$$

where $\hat{Q}_{\mathcal{C}_i^k}(\tau_{\mathcal{C}_i^k}, \mathbf{a}_{\mathcal{C}_i^k}) = \frac{1}{|\mathcal{C}_i^k|} \sum_{j\in\mathcal{C}_i^k} \hat{Q}_j(\tau_j, a_j)$ and $\mathcal{C}_i^k$ is sampled M times from $Pr(\mathcal{C}_i | \mathcal{N}\setminus\{i\})$ (i.e., implemented as Remark 1 suggests) to approximate $\mathbb{E}_{\mathcal{C}_i \sim Pr(\mathcal{C}_i | \mathcal{N}\setminus\{i\})}[\hat{\psi}_i(\mathbf{s}, a_i; \mathbf{a}_{\mathcal{C}_i})]$ using Monte Carlo approximation; and $F_{\mathbf{s}}$ is a monotonic function, followed by an absolute activation function, whose weights are generated from hyper-networks w.r.t. the global state. We show that Eq.12 satisfies the condition to $w_i(\mathbf{s}, a_i)$ in Theorem 1 and therefore the implementation of SHAQ is reliable (see Appendix E.6.1).

By using the framework of fitted Q-learning [24] to solve large number of states (i.e., could be usually infinite) and plugging in the above designed modules, the practical least-square-error loss function

derived from Eq.9 is therefore stated as follows:

$$\min_{\theta,\lambda} \mathbb{E}_{\mathbf{s},\tau,\mathbf{a},R,\tau'}\Big[\Big(R + \gamma \sum_{i\in\mathcal{N}} \max_{a_i} \hat{Q}_i(\tau_i', a_i; \theta^-) - \sum_{i\in\mathcal{N}} \hat{\delta}_i(\mathbf{s}, a_i; \lambda)\, \hat{Q}_i(\tau_i, a_i; \theta) \Big)^2 \Big], \tag{13}$$

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

5 Related Work

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

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

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

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

where subscript $-i$ indicates the agents excluding i. $\bar{Q}_i(\mathbf{s}, a_i)$ can be seen as the action marginal contribution between the grand coalition Q-value and the coalition Q-value excluding the agent i, under *some permutation to form the grand coalition* wherein agent i is located at the *last position*. The efficiency is obviously violated (i.e., the sum of optimal action marginal contributions defined here is unlikely to be equal to the optimal grand coalition Q-value). In contrast to COMA, SHAQ considers all permutations to form the grand coalition to preserve the efficiency.

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

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

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

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

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

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

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

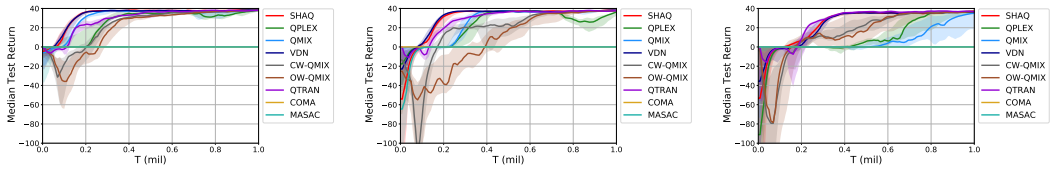
$$\mathbb{E}_{\mathbf{s},\mathbf{a},R,\mathbf{s}'}\Big[\Big(R + \gamma \sum_{i \in \mathcal{N}} \max_{a_i} \hat{Q}_i^{\phi^*}(\mathbf{s}', a_i) - \sum_{i \in \mathcal{N}} \max_{a_i} \hat{Q}_i^{\phi^*}(\mathbf{s}, a_i) \Big)^2 \Big] = 0.$$

However, about suboptimal actions, *SQDDPG does not provide any theoretical guarantee*, whereas SHAQ does with specific implementations as shown in Eq.12 to match the theoretical results shown in this paper. Note that this is critical to reliable interpretations of the optimal MSQ w.r.t. suboptimal actions (e.g., for detecting adversarial attacks on controllers if deployed in industry [34]).

6 Experiments

In this section, we show the experimental results of SHAQ on Predator-Prey [17] and various tasks in StarCraft Multi-Agent Challenge (SMAC) [1]. The baselines that we select for comparison are COMA [7], VDN [8], QMIX [9], MASAC [36], QTRAN [10], QPLEX [37] and W-QMIX (including CW-QMIX and OW-QMIX) [35]. The implementation details of our algorithm are shown in Appendix B.1, whereas the implementation of baselines are from [35] [2]. We also compare SHAQ with SQDDPG [13] [3], which is shown in Appendix C.3. For all experiments, we use the ϵ-greedy exploration strategy, where ϵ is annealed from 1 to 0.05. The annealing time steps vary among different experiments. For Predator-Prey, we apply 1 million time steps for annealing, following the setup from [37]. For the easy and hard maps in SMAC, we apply 50k time steps for annealing, the same as that in [18]; while for the super-hard maps in SMAC, we apply 1 million time steps for annealing to obtain more explorations so that more state-action pairs can be visited. About the replay buffer size, we set as 5000 for all algorithms that is the same as that in [35]. To fairly evaluate all algorithms, we run each experiment with 5 random seeds. All graphs showing experimental results are plotted with the median and 25%-75% quartile shading. The ablation study of SHAQ is shown in Appendix C.4.

6.1 Predator-Prey



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

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

In this section, we run the experiments on a partially-observable task called Predator-Prey [17], wherein 8 predators that we can control aim to capture 8 preys with random policies in a 10x10

[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 [35], the performance is not comparable across versions.

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

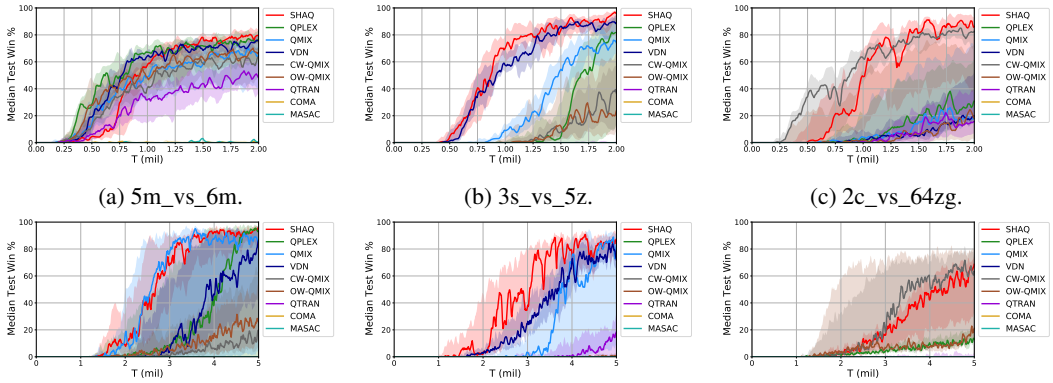
[3] The code of SQDDPG is implemented based on `https://github.com/hsvgbkhgbv/SQDDPG`.

grid world. Each agent's observation is a 5x5 sub-grid centering around it. If a prey is captured by coordination of 2 agents, predators will be rewarded by 10. On the other hand, each unsuccessful attempt by only 1 agent will be punished by a negative reward p. In this experiment, we study the behaviors of each algorithm under different values of p (that describes different levels of coordination). As [35] reported, only QTRAN and W-QMIX can solve this task, while [37] found that the failure was primarily due to the lack of explorations. As a result, we apply the identical epsilon annealing schedule (i.e. 1 million time steps) employed in [37]. As Figure 1 shows, SHAQ can always solve the tasks with different values of p. With the epsilon annealing strategy from [37], W-QMIX does not perform as well as reported in [35]. The reason could be its poor robustness to the increased explorations [35] for this environment (see the evidential experimental results in Appendix C.6). The good performance of VDN validates our analysis in Section 5, whereas the performance of QTRAN is surprisingly almost invariant to the value of p. The performances of QPLEX and QMIX become obviously worse when p=-2. The failure of MASAC and COMA could be due to that relative overgeneralisation[4] prevents policy gradient methods from better coordination [39].

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

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



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

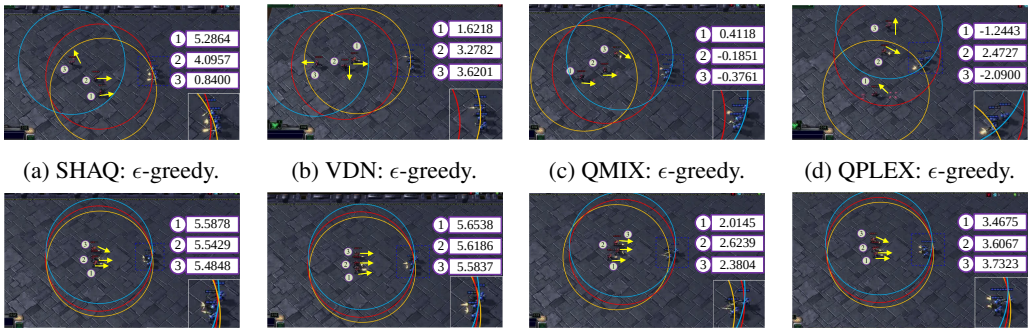
(d) 3s5z_vs_3s6z. (e) Corridor. (f) 6h_vs_8z.

Figure 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), demonstrating the learned MSQs of both ϵ-greedy policy (for obtaining mixed optimal and suboptimal actions) and greedy policy (for obtaining pure optimal actions). As seen from Figure 3a, Agent 3 faces the direction opposite to enemies, meanwhile, the enemies are out of its attacking range. It can be understood as that Agent 3 does not contribute to the team and thus it

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

is almost a dummy agent. Its MSQ is 0.84 (around 0) that correctly catch the manner of a dummy agent (verifying (i) in Proposition 2). In contrast, Agent 1 and Agent 2 are attacking enemies, while Agent 1 suffers from more attacks (with lower health) than Agent 2. As a result, Agent 1 contributes more than Agent 2 and therefore its MSQ is greater, which reflects the credits proportional to the contributions (verifying (iii) in Proposition 2). On the other hand, we can see from Figure 3e that with the optimal policies all agents receive almost identical MSQs (verifying the theoretical results in Section 4.1). The above results well verify the theoretical analysis that we deliver before.



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

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

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

To verify that MSQs learned by SHAQ are non-trivial, we also show the results of VDN, QMIX and QPLEX. It is surprising that the Q-values of these baselines are also almost identical among agents for the optimal actions (however, the property disappears in more complicated scenarios as shown in Appendix C.5 while the property of SHAQ is still valid). Since VDN is a subclass of SHAQ and possesses the same form of loss function for optimal actions, it is reasonable that it obtains the similar results to SHAQ. The explanation for the results of QMIX and QPLEX deserves to be studied in the future work. As for the suboptimal actions, VDN does not possess an explicit interpretation as SHAQ due to the incorrect definition of $\delta_i(\mathbf{s}, a_i) = 1$ over suboptimal actions (verifying the statement in Section 5). Similarly, QMIX and QPLEX cannot show explicit interpretations over suboptimal actions either. To enable the conclusion to be more convincing, we also visualise the results of SHAQ on a more complicated scenario, 3s5z_vs_3s6z (see Appendix C.5).

7 Conclusion

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

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Checklist

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

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

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