VAAC: V-value Attention Actor-Critic for Cooperative Multi-agent Reinforcement Learning

Haonan Liu, Liansheng Zhuang(✉), Yihong Huang, and Cheng Zhao

University of Science and Technology of China, Hefei 230027, China
phoenix_@mail.ustc.edu.cn, lszhuang@ustc.edu.cn

Abstract. This paper explores value-decomposition methods in cooperative multi-agent reinforcement learning (MARL) under the paradigm of centralized training with decentralized execution. These methods decompose a global shared value into individual ones to guide the learning of decentralized policies. While Q-value decomposition methods such as QMIX show state-of-the-art performance, V-value decomposition methods are proposed to obtain a reasonable trade-off between training efficiency and algorithm performance under the A2C training paradigm. However, existing V-value decomposition methods lack theoretical analysis of the relation between the global V-value and local V-values, and do not explicitly consider the influence of individuals on the total system, which degrades their performance. To address these problems, this paper proposes a novel approach called V-value Attention Actor-Critic (VAAC) for cooperative MARL. We theoretically derive a general decomposing formulation of the global V-value in terms of local V-values of individual agents, and implement it with a multi-head attention formation to model the impact of individuals on the whole system for interpretability of decomposition. Evaluations on the challenging StarCraft II micromanagement task show that VAAC achieves a better trade-off between training efficiency and algorithm performance, and provides interpretability for its decomposition process.

Keywords: Multi-agent reinforcement learning · Multi-agent policy gradients · Deep reinforcement learning

1 Introduction

Cooperative multi-agent reinforcement learning (MARL) has made significant progress in recent years [7,15,16], where a system of agents learns towards coordinated policies to optimize the accumulated global rewards. Many complex real-world tasks such as autonomous vehicle coordination [1] and sensor networks [20] can be modeled as cooperative MARL problems. One natural way to address cooperative MARL problems is the fully centralized approach that views the multi-agent system as a single-agent reinforcement learning task with

M. Tanveer et al. (Eds.): ICONIP 2022, LNCS 13623, pp. 562–573, 2023.
https://doi.org/10.1007/978-3-031-30105-6_47

a joint action space [13]. However, since the joint action space of the agents grows exponentially with the number of agents, the fully centralized approach has limited scalability. Besides, due to partial observability and communication constraint in practical environments, it is necessary to use decentralized policies that act only based on the local observation history of individual agents. The simplest approach to decentralized policies is independent learning which trains the agents independently, but suffers from non-stationarity since it views other agents as part of the environment [5].

To address these above issues, the paradigm of *centralized training with decentralized execution* (CTDE) [4] has attracted great attention of researchers, where decentralized policies are trained with access to additional global state information in a centralized fashion and executed only conditioned on local histories in a decentralized way. There is still a challenging problem of how to use a global shared value, such as joint action-value (Q-value) or global state value (V-value), for the training of decentralized policies. One approach is value-decomposition, which decomposes the global value into individual ones to guide the learning of decentralized policies. Many breakthroughs in Q-value decomposition methods have been made recently. *Value Decomposition Network* (VDN) [12] represents joint action-value Q_{tot} as a summation of individual Q-values that condition only on individual observations and actions. QMIX [8] extends to a broader class of monotonic functions using a mixing network of per-agent Q-values. QTRAN [10] proposes a provably more general value factorization method that avoids representation limitations. Qatten [18] theoretically derives a general formula of Q_{tot} and considers the impact of individuals on the global.

In real applications, how to improve the training efficiency of CTDE is a practical problem in cooperative MARL. A2C framework [6] is a popular training paradigm that promotes training efficiency by asynchronously executing multiple instances of the environment. However, Q-value decomposition methods such as QMIX do not perform well under the A2C paradigm, because these off-policy methods utilize replay buffers which are incompatible with multi-thread execution. As reported in [11], when using the A2C training paradigm, the performance of QMIX degrades on the StarCraft Multi-Agent Challenge (SMAC) [9]. On the other hand, on-policy actor-critic methods such as *counterfactual multi-agent* (COMA) [2] can exploit the A2C framework efficiently while having poor performance on SMAC [9].

To narrow the performance gap between on-policy actor-critic and Q-value decomposition methods, [11] extends value-decomposition to on-policy actor-critic methods and proposes a V-value decomposition framework called *value-decomposition actor-critic* (VDAC). VDAC represents the global state value V_{tot} as a monotonic function of local state values V^a, and introduces two V-value decomposition methods, i.e. VDAC-sum and VDAC-mix. The former method represents V_{tot} as a summation of V^a, while the later one generalizes the representation to a larger family of monotonic functions through a mixing network. However, both V-value decomposition methods impose certain assumptions which lack theoretical analysis of the relation between V_{tot} and V^a. Besides,

they do not explicitly consider the influence of individuals on the total system, just viewing that each agent is equal or mixing local state values implicitly. These problems of existing V-value decomposition methods limit their performance.

To further achieve an acceptable trade-off between training efficiency and performance, this paper proposes a novel V-value decomposition approach called V-value Attention Actor-Critic (VAAC). We derive a decomposing formulation of V_{tot} in terms of V^a through theoretical analysis, and implement it with a multi-head attention formation to model the impact of agents on the whole system for interpretability of decomposition. Empirical results on SMAC show that VAAC outperforms other baselines under A2C. Next, we use ablation experiments to demonstrate the contribution of the multi-head attention formation. Moreover, we investigate the relationship between the weights for mixing V^a into V_{tot} and the properties of agents to interpret the decomposition process.

2 Background

2.1 Decentralized Partially Observable Markov Decision Process

We consider a fully cooperative multi-agent task that can be modeled as a *decentralised partially observable Markov decision process* (Dec-POMDP) consisting of a tuple $G = \langle S, U, P, r, Z, O, n, \gamma \rangle$, in which n agents identified by $a \in A \equiv \{1, ..., n\}$ choose sequential actions. The environment has a true state $s \in S$. Each agent simultaneously chooses an action $u^a \in U$ at each time step, forming a joint action $\mathbf{u} \in \mathbf{U} \equiv U^n$ which induces a transition probability function $P(s'|s, \mathbf{u}) : S \times \mathbf{U} \times S \rightarrow [0, 1]$ and a global reward function $r(s, \mathbf{u}) : S \times U \rightarrow \mathbb{R}$. We consider a partially observable setting, where each agent receives an individual partial observation $z \in Z$ from the observation function $O(S, A) : S \times A \rightarrow Z$. Each agent learns a stochastic policy $\pi^a(u^a|\tau^a) : T \times U \rightarrow [0, 1]$ conditioned on its local action-observation history $\tau^a \in T \equiv Z \times U$. We denote joint quantities over agents in bold and joint quantities over agents other than a given agent a with the superscript $-a$. All agents coordinate together to maximize the discounted return $R_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}$. The agents' joint policy induces a value function, i.e., the expected return for following the joint policy π from state s, $V^\pi(s_t) = \mathbb{E}[R_t|s_t = s]$, and an action-value function, i.e. the expected return for selecting joint action $\mathbf{u}$ in state s and following the joint policy π, $Q^\pi(s, \mathbf{u}) = \mathbb{E}[R_t|s_t = s, \mathbf{u}]$.

2.2 Single-Agent Policy Gradient Algorithms

Policy gradient methods optimise a single agent's policy parameterised by θ_π to maximize the objective $J(\theta) = \mathbb{E}_{s \sim p^\pi, u \sim \pi}[R(s, u)]$ by performing gradient ascent, where p^π is the state transition by following policy π. The gradient with respect to the policy parameters is $\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi_\theta(a|s) Q_\pi(s, u)]$. To reduce variations in gradient estimates, a baseline b is introduced. In *actor-critic* approaches, the actor, i.e., the policy, is trained by following a gradient that

depends on a critic, which usually estimates a value function. This yields the advantage function $A(s_t, u_t) = Q(s_t, u_t) - b(s_t)$. $V(s_t)$ is commonly used as the baseline. Temporal difference (TD) error $r_t + \gamma V(s_{t+1}) - V(s_t)$, which is an unbiased estimate of $A(s_t, u_t)$, is a common choice for advantage functions.

2.3 IAC and COMA

Independent Actor-Critic (IAC) [2] is the simplest method to apply Policy Gradient Algorithms to multiple agents, which lets each agent learn its own actor and critic independently according to its own action-observation history. Each agent's critic estimates $V(o^a)$ to calculate TD error. IAC is straightforward and easy to implement but lacks information about other agents and global state during the training, which makes it difficult to learn coordinated strategies and estimate its contribution to the team's reward. To mitigate this issue, decentralized policies can be learned in the centralized training and decentralized execution (CTDE) paradigm. COMA [2] uses a centralized critic and applies the following counterfactual policy gradients:$\nabla_\theta J = \mathbb{E}_\pi \left[\sum_a \nabla_\theta \log \pi(u^a|\tau^a) A^a(s, \mathbf{u})\right]$, where $A^a(s, \mathbf{u}) = Q_\pi(s, \mathbf{u}) - \sum_{u^a} \pi_\theta(u^a|\tau^a) Q^a_\pi(s, (\mathbf{u}^{-a}, u^a))$ is the counterfactual advantage for agent a. COMA provides agents with tailored gradients to achieve credit assignment, but it becomes ineffective with complex cooperation behaviors.

2.4 Value Decomposition Actor-Critic

Value Decomposition Actor-Critic (VDAC) [11] is an actor-critic, on-policy framework that uses the paradigm of CTDE. VDAC has local critics for each agent to estimate the local state values V^a and a central critic to estimate the global state value V_{tot}. Inspired by *difference rewards* [17], VDAC decomposes the global state value $V_{tot}(s)$ into local states $V^a(o^a)$ through the following constraint:

$$\frac{\partial V_{tot}}{\partial V^a} \geq 0, \qquad \forall a \in \{1, ..., n\}. \tag{1}$$

With Eq. 1 enforced, given that the other agents stay at the same local states by taking $\mathbf{u}^{-a}$, any action u^a that leads agent a to a local state o^a with a higher value will also improve the global state value V_{tot}. In [11], they also prove the convergence of VDAC frameworks to a locally optimal policy. Two variants of value-decomposition that satisfy Eq. 1, VDAC-sum and VDAC-mix, are proposed in [11]. In VDAC-sum, $V_{tot}(s)$ is represented by a summation of local state values V^a, $V_{tot}(s) = \sum_a V^a(o^a)$. This linear representation satisfies the constraint. θ denotes the actors' parameters and θ_v denotes the distributed critics' parameters. The distributed critic is optimized by minibatch gradient descent to minimize the following loss $L_t(\theta_v) = (y_t - \sum_a V_{\theta_v}(o^a_t))^2$, where $y_t = \sum_{i=t}^{k-t-1} \gamma^i r_i + \gamma^{k-t} V_{tot}(s_k)$ is bootstrapped from the last state s_k, and k is upper-bounded by T. To generalize the representation to a larger

family of monotonic functions, VDAC-mix uses a feed-forward neural network that takes input as local state values $V^a(o^a), \forall a \in \{1, ..., n\}$, and outputs the global state value V_{tot}. To enforce Eq. 1, the weights (not including bias) of the mixing network, which are produced by separate hypernetworks [3] that take s as an input, are restricted to be non-negative. The distributed critics are optimized by minibatch gradient descent to minimize the following loss $L_t(\theta_v) = (y_t - V_{tot}(s_t))^2 = (y_t - f_{mix}(V_{\theta_v}(o_t^1), ..., V_{\theta_v}(o_t^n)))^2$, where f_{mix} denotes the mixing network. The central critic is optimized by minimizing the same loss $L_t(\theta^c) = (y_t - V_{tot}(s_t))$, where θ^c denotes parameters in the hypernetworks. The policy network is trained using the following policy gradient $g = \mathbb{E}_\pi \left[\sum_a \nabla_\theta \log \pi^a(u^a|\tau^a) A(s, \mathbf{u})\right]$, where $A(s, \mathbf{u}) = r_t + \gamma V(s') - V(s)$ is a simple TD advantage.

3 Method

In this section, we propose a new V-value decomposition approach called V-value Attention Actor-Critic (VAAC). First, we perform the theoretical analysis of global and local V-values and derive a general decomposition formula. Then, we describe the architecture of VAAC that uses a multi-head attention formation to implement the decomposition formula.

3.1 Theoretical Analysis

Considering a stochastic policy π, the relationship between $V^\pi(s)$ and $Q^\pi(s, u)$ is $V^\pi(s) = \sum_u \pi(a|s) Q^\pi(s, u)$. Thus, V_{tot} and V^a can be formulated as:

$$V_{tot}(s) = \sum_{\mathbf{u}} \pi(\mathbf{u}|s) Q_{tot}(s, \mathbf{u}), V^a(o^a) = \sum_{u^a} \pi^a(u^a|\tau^a) Q^a(o^a, u^a), \tag{2}$$

where π denotes the joint policy and π^a denotes the individual policy of agent a. In [18], they theoretically derive a general decomposition formula of Q_{tot} by local state-action values Q^a:

$$Q_{tot}(s, \mathbf{u}) \approx c(s) + \sum_{a,h} \lambda_{a,h}(s) Q^a(o^a, u^a), \tag{3}$$

where $\mathbf{u} = (u^1, ..., u^n)$ and $\lambda_{a,h}$ is a linear functional of all partial derivatives $\frac{\partial^h Q_{tot}}{\partial Q^{a_1} ... \partial Q^{a_h}}$ of order h, decaying super exponentially fast in h. Equation 3 appears to be a linear relationship between Q_{tot} and Q^a, yet contains the non-linear information due to the coefficient $\lambda_{a,h}$ that is a function of all partial derivatives of order h, and corresponds to all cross-terms $Q^{a_1} ... Q^{a_h}$ of order h. Therefore, we could decompose V_{tot} to V^a as shown in the following Theorem 1.

Theorem 1. *Assuming that the joint policy π can be formulated as a product of independent actors: $\pi(\mathbf{u}|s) = \Pi_a \pi^a(u^a|\tau^a)$, then*

$$V_{tot}(s) \approx c(s) + \sum_{a,h} \lambda_{a,h}(s) V^a(o^a) \tag{4}$$

where c and $\lambda_{a,h}$ depend on the global state s, $\lambda_{a,h}$ is a linear functional of all partial derivatives $\frac{\partial^h Q_{tot}}{\partial Q^{a_1} \ldots \partial Q^{a_h}}$ of order h, decaying super exponentially fast in h.

Proof. Since $\pi(\mathbf{u}|s) = \Pi_a \pi^a(u^a|\tau^a)$, we have $\pi(\mathbf{u}|s) = \pi(\mathbf{u}^{-a}|\tau^{-a})\pi^a(u^a|\tau^a)$, where $\pi(\mathbf{u}^{-a}|\tau^{-a})$ denotes a product of independent actors other than a given agent a, according to Eqs. 2 and 3 we have:

$$\begin{aligned}
V_{tot}(s) &\approx \sum_{\mathbf{u}} \pi(\mathbf{u}|s)\Big(c(s) + \sum_{a,h} \lambda_{a,h}(s) Q^a(o^a, u^a)\Big) \\
&= c(s) \sum_{\mathbf{u}} \pi(\mathbf{u}|s) + \sum_{\mathbf{u}} \pi(\mathbf{u}|s) \sum_{a,h} \lambda_{a,h}(s) Q^a(o^a, u^a) \\
&= c(s) + \sum_{a,h} \lambda_{a,h}(s)\Big(\sum_{\mathbf{u}^{-a}} \pi(\mathbf{u}^{-a}|\tau^{-a}) \sum_{u^a} \pi^a(u^a|\tau^a) Q^a(o^a, u^a)\Big) \\
&= c(s) + \sum_{a,h} \lambda_{a,h}(s)\Big(\sum_{\mathbf{u}^{-a}} \pi^a(\mathbf{u}^{-a}|\tau^{-a}) V^a(o^a)\Big) \\
&= c(s) + \sum_{a,h} \lambda_{a,h}(s)\Big(V^a(o^a) \sum_{\mathbf{u}^{-a}} \pi^a(\mathbf{u}^{-a}|\tau^{-a})\Big) \\
&= c(s) + \sum_{a,h} \lambda_{a,h}(s) V^a(o^a)
\end{aligned} \tag{5}$$

3.2 Implementation

Following the above decomposition formula in Eq. 4, we propose VAAC based on the attention mechanism [14]. Figure 1 illustrates the overall architecture of VAAC. For each agent a, there is one agent network, which receives its action-observation history τ^a (last hidden states h^a_{t-1} and current local observation o^i_t) and outputs both $\pi^a(o^a)$ and $V^a(o^a)$ by sharing non-output layers between distributed critics and actors. The key to the mixing process is how to approximate different weights $\lambda_{a,h}$ corresponding to agent a and order h in Eq. 4. Thus, following the practice in Qatten [18], we feed local state values V^a and additional global state information (including the global state s and the agent's individual features μ^a) into the mixing network using a multi-head attention formation to model the individual impacts.

In Eq. 4, let the outer sum over h:

$$V_{tot}(s) \approx c(s) + \sum_{h=1}^{H} \sum_{a=1}^{N} \lambda_{a,h}(s) V^a(o^a). \tag{6}$$

First, for each h, the inner weighted sum operation can be implemented by the differentiable key-value memory model [19], we compute the weights $\lambda_{a,h}$ as:

$$\lambda_{a,h} \propto softmax\Big(\frac{e_a^T e_s}{\sqrt{d}}\Big), \tag{7}$$

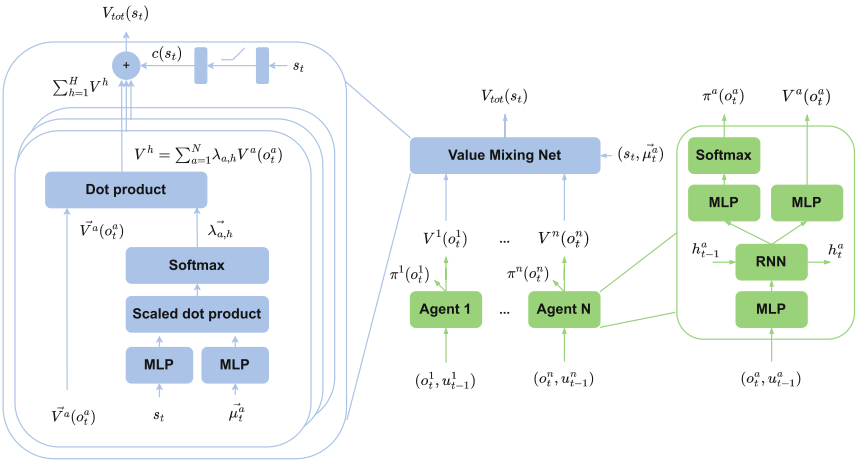


Fig. 1. The overall architecture of VAAC.

where e_s and e_a are obtained by a two-layer embedding transformation for s and μ^a, d is the embedding dim. Note that individual features μ^a are the part of the global state s related to agent a, hence $\lambda_{a,h}$ still only depends on s when we compute in Eq. 7. Then we compute the weighted sum of the local values V^a as $V^h = \sum \lambda_{a,h} V^a(o^a)$, where V^h denotes the output of a single attention. Next, for the outer sum over h, we adopt multiple attention heads to correspond to different orders of partial derivatives. Since $\lambda_{a,h}$ decays super exponentially fast in h, we stop at H for the feasibility of implementation, where H denotes the number of attention heads. Adding up the outputs of different heads and $c(s)$ which is produced by a two-layer network with the global state s as the input, we have

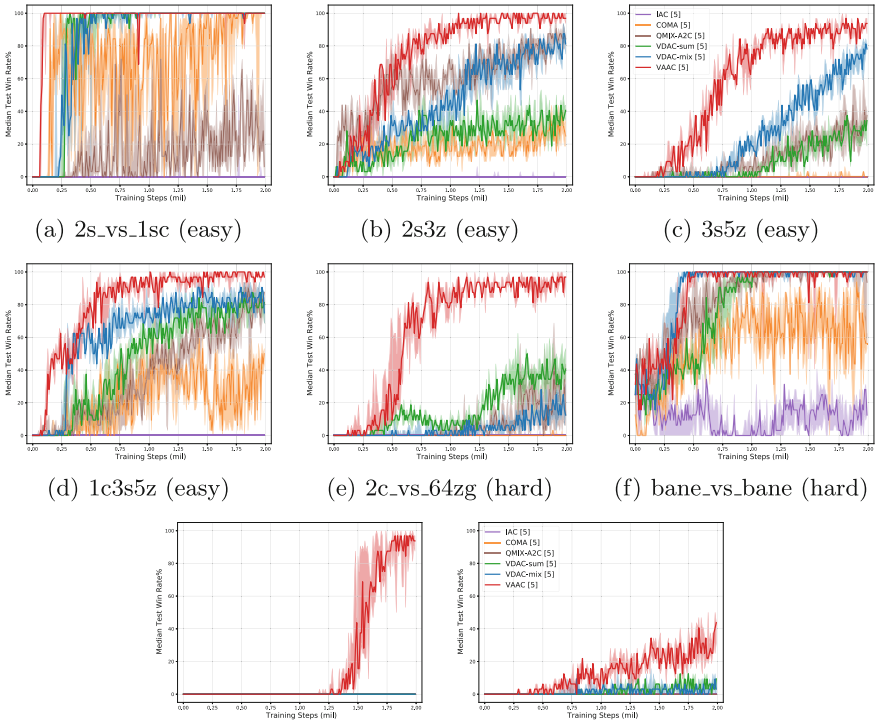
$$V_{tot}(s) \approx c(s) + \sum_{h=1}^{H} V^h. \tag{8}$$

Naturally, VAAC satisfies the constraint of VDAC in Eq. 1. The distributed critics and the value mixing network are optimized by minibatch gradient descent to minimize the following loss $L_t(\theta) = (y_t - V_{tot}(s_t))^2$. The policy network is trained using the following policy gradient $g = \mathbb{E}_\pi \left[\sum_a \nabla_\theta \log \pi^a(u^a|\tau^a) A(s, \mathbf{u})\right]$, where $A(s, \mathbf{u}) = r_t + \gamma V_{tot}(s_{t+1}) - V_{tot}(s_t)$ is a simple TD advantage.

4 Experiments

We evaluate VAAC against previous state-of-the-art multi-agent on-policy actor-critic methods such as VDAC-sum [12], VDAC-mix [12], COMA [2] and IAC [2], and multi-agent Q-learning method QMIX under A2C training paradigm (QMIX-A2C) [12] on the StarCraft Multi-Agent Challenge (SMAC) environment [9], in which each agent controls an individual allied army unit to beat

the enemy. SMAC consists of various maps which have been classified as *easy*, *hard*, and *super hard*. We consider the following maps in our experiments: four *easy* maps (2s_vs_1sc, 2s3z, 3s5z, and 1c3s5z), three *hard* maps (2c_vs_64zg, bane_vs_bane and 3s_vs_5z) and one *super hard* map (MMM2). Note that all algorithms are trained under the A2C training paradigm where 8 episodes are rolled out independently during the training. Our method uses RMSprop with learning rate 0.0025, γ is set to 0.99 and λ is set to 0.8. For baseline algorithms, we use the same training setup as provided by their authors. The agent networks resemble a DRQN with a recurrent layer comprised of a GRU with a 64-dimensional hidden state, with a fully-connected layer before and after. The agent networks contain an additional layer to output local state values and the policy network outputs a stochastic policy. $c(s)$ is produced by a two-layer network with a 32-dimensional hidden state. We use ReLU for all activation functions. For the attention part, query (global state s) and key (agent's individual features μ^a) are obtained by two-layer embedding transformations, where hidden state dim is 64 and embedding dim is 32, and the number of heads H is 4.



(a) 2s_vs_1sc (easy) (b) 2s3z (easy) (c) 3s5z (easy)

(d) 1c3s5z (easy) (e) 2c_vs_64zg (hard) (f) bane_vs_bane (hard)

(g) 3s_vs_5z (hard) (h) MMM2 (super hard)

Fig. 2. Median win percentage on eight different SMAC maps

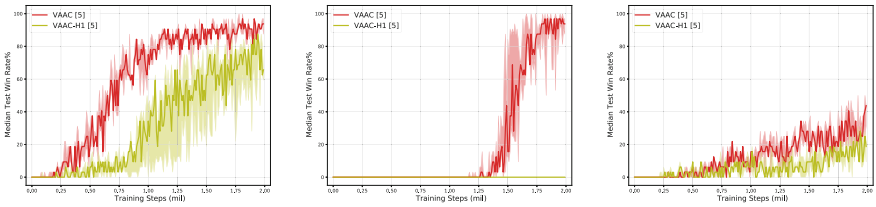
4.1 Main Results

We compare VAAC with other baselines on all maps mentioned above. The main evaluation metric is the median win percentage of evaluation episodes as a function of environment steps observed over the 2 million training steps [9]. The training is paused after every 10000 timesteps during which 32 test episodes are run with agents performing action selection greedily in a decentralized fashion. The median performance as well as the 25-75% percentiles are obtained by 5 independent training runs with different seeds. Figure 2 presents results.

In all scenarios, VAAC outperforms other baselines under A2C. On the *easy* and *hard* maps, VAAC can master these tasks and achieve competitive performance. Even on the *super hard* map, VAAC's win percentage can reach approximately 40%. VDAC-mix has good performance on *easy* maps but performs not well on *hard* and *super hard* maps due to the difficulty and complexity.

4.2 Ablations

We perform ablation experiments to investigate the influence and necessity of the multi-head attention formation. In our implementation, we use multiple attention heads to approximate different orders of partial derivatives in Eq. 4. Because $\lambda_{a,h}$ decays super exponentially fast in h, we stop at $H = 4$ which denotes the number of attention heads. We compare against VAAC without the multi-head attention formation by setting $H = 1$. We refer to this method as VAAC-H1. We test on the 3s5z (*easy*), 3s_vs_5z (*hard*), and MMM2 (*super hard*) maps. Figure 3 shows that VAAC outperforms VAAC-H1. It reveals that the multi-head attention formation could capture sophisticated relations between V_{tot} and V^a to improve performance.



(a) 3s5z (easy) (b) 3s_vs_5z (hard) (c) MMM2 (super hard)

Fig. 3. Ablations of VAAC on three SMAC maps

4.3 Weights Analysis

We analyze the attention weights $\lambda_{a,h}$ to show how our method models the individual impact of agents, which provides interpretability for our method's decomposition process. In our implementation, the attention weights $\lambda_{a,h}$ are

produced by the global state s, agent's individual features μ^a, and corresponding to different orders using multiple heads. We choose the 1c3s5z map as the representative for experiments since it contains three different types of units (including 1 Colossi, 3 Stalkers, and 5 Zealots).

First, we calculate the mean of $\lambda_{a,h}$ for different agents and heads. For clarity, we consider the difference between the mean of $\lambda_{a,h}$ and the average weight (Specifically, on the 1c3s5z map with nine agents, the average weight is $\frac{1}{9}$). Figure 4 presents the results. For each agent, the weights $\lambda_{a,h}$ of different heads are not the same, but in general, the higher value unit types of agents have higher weights. As shown in Fig. 4, Colossi's weights are much higher than others, the weights of Stalkers are roughly around the average weight and Zealots have the lowest weights. It demonstrates that our method could assign higher weights to more powerful and impactful agents.

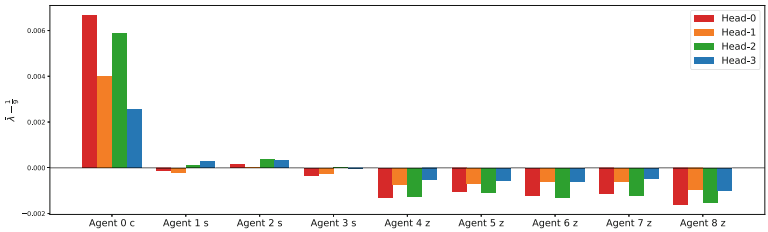
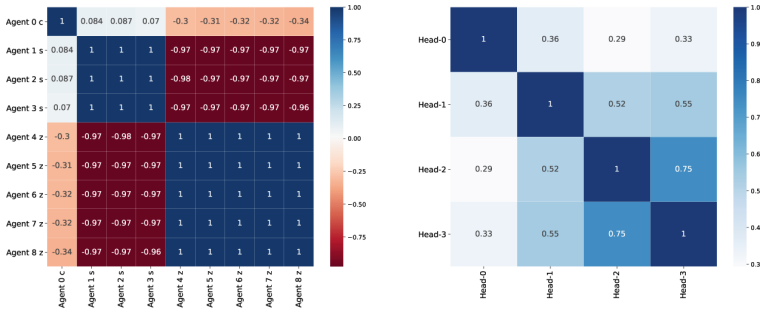


Fig. 4. The difference between the mean of $\lambda_{a,h}$ and the average weight (1/9 on this map) for different agents and heads on 1c3s5z map

Next, since the attention weights $\lambda_{a,h}$ are adaptive to the global state, we calculate the matrix of correlation coefficients for $\lambda_{a,h}$ during the training from the perspective of agents and attention heads respectively. Figure 5(a) shows the matrix of correlation coefficients from the perspective of Head-0. We find a strong positive correlation between the weights of agents of the same type, such as Stalkers (Agent 1, 2, and 3) or Zealots (Agent 4-8). Agents of the same type tend to perform the same function as a group. Interestingly, we notice that the weights of Stalkers and Zealots are strongly negatively correlated. For the 1c3s5z map, Stalkers and Zealots should maintain the formation to protect the Colossi that can deal a lot of damage to the enemy. Due to cooldown or other features, Stalkers and Zealots take turns taking a more important role in the team, reflected in the ebb and flow of their weights according to the team's needs. Figure 5(b) shows the matrix of correlation coefficients from the perspective of Agent 0. Generally, for a given agent, there is a certain degree of positive correlation between the weights $\lambda_{a,h}$ of different heads. According to Fig. 4 and 5(b), different attention heads may capture different features in sub-spaces to approximate the weights of different orders.



(a) For Head-0 (b) For Agent 0 c

Fig. 5. The matrix of correlation coefficients for $\lambda_{a,h}$ on 1c3s5z map

5 Conclusion

In this paper, we propose V-value Attention Actor-Critic (VAAC) for cooperative MARL. First, we derive a decomposition formulation of the global state value V_{tot} in terms of local state values V^a. To implement the decomposition formula, we adopt the multi-head attention formation in the mixing network, which can explicitly model the impact of individuals on the whole system for interpretability of decomposition. Experiments on the StarCraft II micromanagement task demonstrate that our method VAAC not only reaches better performance, but also provides interpretability for its decomposition process. In future work, we aim to further research on V-value decomposition methods.

Acknowledgement. This work was supported in part to Dr. Liansheng Zhuang by NSFC under contract No.U20B2070 and No.61976199.

References

1. Cao, Y., Yu, W., Ren, W., Chen, G.: An overview of recent progress in the study of distributed multi-agent coordination. IEEE Trans. Industr. Inf. **9**(1), 427–438 (2012)
2. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
3. Ha, D., Dai, A., Le, Q.V.: Hypernetworks. arXiv preprint arXiv:1609.09106 (2016)
4. Kraemer, L., Banerjee, B.: Multi-agent reinforcement learning as a rehearsal for decentralized planning. Neurocomputing **190**, 82–94 (2016)
5. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems 30 (2017)
6. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
7. Qiu, W., et al.: RMIX: learning risk-sensitive policies for cooperative reinforcement learning agents. In: Advances in Neural Information Processing Systems 34 (2021)

8. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S.: QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 4295–4304 (2018)
9. Samvelyan, M., et al.: The starcraft multi-agent challenge. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2186–2188 (2019)
10. Son, K., Kim, D., Kang, W.J., Hostallero, D.E., Yi, Y.: QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 5887–5896 (2019)
11. Su, J., Adams, S., Beling, P.A.: Value-decomposition multi-agent actor-critics. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11352–11360 (2021)
12. Sunehag, P., et al.: Value-decomposition networks for cooperative multi-agent learning based on team reward. In: AAMAS (2018)
13. Tan, M.: Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the Tenth International Conference on Machine Learning, pp. 330–337 (1993)
14. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (2017)
15. Wang, J., Ren, Z., Liu, T., Yu, Y., Zhang, C.: QPLEX: duplex dueling multi-agent Q-learning. In: International Conference on Learning Representations (2020)
16. Wang, T., Gupta, T., Peng, B., Mahajan, A., Whiteson, S., Zhang, C.: Rode: learning roles to decompose multi-agent tasks. In: Proceedings of the International Conference on Learning Representations (2021)
17. Wolpert, D.H., Tumer, K.: Optimal payoff functions for members of collectives. In: Modeling Complexity in Economic and Social Systems, pp. 355–369. World Scientific (2002)
18. Yang, Y., et al.: Qatten: a general framework for cooperative multiagent reinforcement learning. arXiv preprint arXiv:2002.03939 (2020)
19. Yun, C., Bhojanapalli, S., Rawat, A.S., Reddi, S., Kumar, S.: Are transformers universal approximators of sequence-to-sequence functions? In: International Conference on Learning Representations (2019)
20. Zhang, C., Lesser, V.: Coordinated multi-agent reinforcement learning in networked distributed POMDPs. In: Twenty-Fifth AAAI Conference on Artificial Intelligence (2011)