NA^{2}Q: Neural Attention Additive Model for Interpretable Multi-Agent Q-Learning

Zichuan Liu [1] **Yuanyang Zhu** [1] **Chunlin Chen** [1]

Abstract

Value decomposition is widely used in cooperative multi-agent reinforcement learning, however, its implicit credit assignment mechanism is not yet fully understood due to black-box networks. In this work, we study an interpretable value decomposition framework via the family of generalized additive models. We present a novel method, named Neural Attention Additive Q-learning (NA^{2}Q), providing inherent intelligibility of collaboration behavior. NA^{2}Q can explicitly factorize the optimal joint policy induced by enriching shape functions to model all possible coalitions of agents into individual policies. Moreover, we construct identity semantics to promote estimating credits together with the global state and individual value functions, where local semantic masks help us diagnose whether each agent captures relevant-task information. Extensive experiments show that NA^{2}Q consistently achieves superior performance compared to different state-of-the-art methods on all challenging tasks, while yielding human-like interpretability.

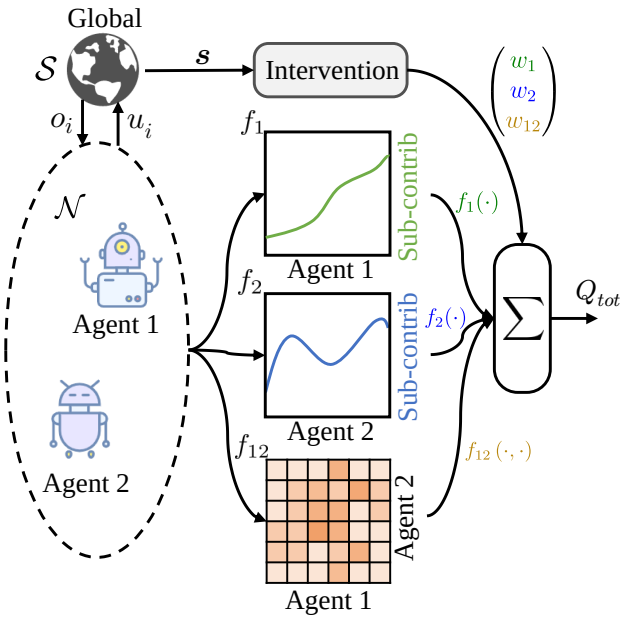


Figure 1. An example of value decomposition via the GAMs family in MARL, where $s \in \mathcal{S}$ is the global state, $f_k \in \{f_1, \cdots, f_{1\ldots n}\}$ denotes the contribution of a shape function to learning individual or pairwise action values, and Q_{tot} denotes the joint action value.

1. Introduction

Cooperative multi-agent reinforcement learning (MARL) has been proven to hold considerable promise for addressing many challenging real-world problems, e.g., autonomous driving (Kiran et al., 2021), scene understanding (Chen et al., 2019), and robotics (Kober et al., 2013; Lillicrap et al., 2016). Value decomposition (Rashid et al., 2018; Son et al., 2019; Wang et al., 2021) has witnessed success in handling the joint action-value function effectively in value-based MARL methods. This progress has been fueled by black-box neural structures, where the underlying decision process and credit assignment mechanisms are difficult for humans to understand and interpret. Hence, explicitly understanding the decision-making processes and deducing the contribution of agents is still crucial in the MARL community.

A growing body of work attempts to demystify the decision-making process of deep reinforcement learning. Instance-specific approximation methods aim to explain black-box predictions via the Shapley value (Wang et al., 2020) or clustering (Zahavy et al., 2016) techniques in post-hoc explanation techniques. However, these interpretable methods are considered computationally expensive (Slack et al., 2021) and unstable (Ghorbani et al., 2019), i.e., they often misrepresent models or agents' decisions. Other works (Bastani et al., 2018; Silva et al., 2020) have resorted to imitation learning to generate post-hoc global explanations aimed at distilling agent strategies, which lack the transparency of the original model and do not guarantee performance in

[1]Department of Control Science and Intelligence Engineering, Nanjing University, Nanjing, China. Correspondence to: Yuanyang Zhu <yuanyang@smail.nju.edu.cn>, Chunlin Chen <clchen@nju.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

complex tasks. This landscape has ignited interest in intrinsic explanations, particularly in generalized additive models (GAMs) (Hastie & Tibshirani, 1986). GAMs typically learn independent shape functions for each feature, whose outputs are combined for the final prediction, ensuring isolated contributions, e.g., NIT (Tsang et al., 2018), NAM (Agarwal et al., 2021), and NODE-GAM (Chang et al., 2022). Moreover, it can model all higher-order feature interactions with expressive power and easy scalability. These successful interpretable GAMs stimulate our thinking in MARL domains, i.e., could GAMs facilitate more trustworthy agent collaboration and efficient credit assignment?

To leverage the benefits of GAM in MARL effectively, we introduce a unique value decomposition workflow as illustrated in in *Figure 1*. At each timestep t, each agent learns the decentralized action-value Q_i and passes it to the central GAM while executing the action u_i, which then evaluates the team utility via the joint action-value Q_{tot}. Specifically, our GAM consists of several independent shape functions, where each function inputs a marginal or higher-order action value, outputting the corresponding agent's team contribution. We restrict our attention to unary and pairwise shape functions to maintain interpretability and efficiently infer credits, helping in isolating individual and pairwise coalition contributions. However, the causal confounder is correlated with $\boldsymbol{s}$ and Q_{tot}, creating a spurious correlation among them, potentially complicating the learning of correct causal relationships. Drawing inspiration by (Glymour et al., 2016; Li et al., 2022a), to relieve the spurious correlation between $\boldsymbol{s}$ and Q_{tot}, we construct local semantics alongside the global state to compute credits. In this case, it explicitly provides a perspective on diagnosing whether the individual agent could effectively avoid the negative influence of focusing on irrelevant input information. Meanwhile, this brings about semantic masks that can diagnose agents' local observations. We utilize the attention mechanism (Vaswani et al., 2017) as an intervention term to capture the credit of each shape function, facilitating effectively capturing credit assignment. We call this comprehensive solution Neural Attention Additive Q-learning (NA^{2}Q), which offers a fresh perspective for interpreting collaboration among agents and understanding local semantics.

Our contributions are summarized as follows: (1) We propose a novel value decomposition method, called Neural Attention Additive Q-learning (NA^{2}Q), which moves a step towards modeling all possible higher-order interactions and interpreting their collaboration behavior. We give rigorous proof that NA^{2}Q guarantees an acceptable regret bound by enriching the Taylor expansion of Q_{tot} based on the GAM family. (2) We provide diagnostic insights into what the agent captured from its observation by maximizing the observation resemblance and generating masks through encoding the local semantics, which is applied to the mixer to promote credit deduction. (3) Through extensive experiments on challenging MARL benchmarks, NA^{2}Q not only consistently achieves superior performance compared to different state-of-the-art methods but also allows for an easy-to-understand of credit assignment among agents.

2. Preliminaries

2.1. Dec-POMDP

A fully cooperative multi-agent task generally can be formulated as a Dec-POMDP (Oliehoek & Amato, 2016), which consists of a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{U}, \mathcal{P}, r, O, \Omega, \gamma \rangle$, where $\mathcal{N}$ represents a finite set of n agents, and $\boldsymbol{s} \in \mathcal{S}$ describes the global state of the environment. At each time step, each agent $i \in \mathcal{N}$ receives its own observation $o_i \in \Omega$ according to the partial observation $O(\boldsymbol{s}, i)$ and chooses an action $u_i \in \mathcal{U}$ to formulate a joint action $\boldsymbol{u} = [u_i]_{i=1}^n \in \mathcal{U}^n$. It results in a next state transition $\boldsymbol{s}'$ according to the transition function $\mathcal{P}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{u}) : \mathcal{S} \times \mathcal{U}^n \to \mathcal{S}$ and all agents receive a joint reward $r(\boldsymbol{s}, \boldsymbol{u}) : \mathcal{S} \times \mathcal{U}^n \to \mathbb{R}$. Moreover, each agent i learns its own policy $\pi_i(u_i|\tau_i) : \mathcal{T} \times \mathcal{U} \to [0, 1]$ conditions on its local action-observation history $\tau_i \in \mathcal{T}$, and we define $\boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$ to denote joint action-observation history. The formal goal of all agents is to maximize the joint value function $Q^{\boldsymbol{\pi}} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^t\right]$ that finds an optimal joint policy $\boldsymbol{\pi} = [\pi_i]_{i=1}^n$, where $\gamma \in [0, 1)$ is a discount factor.

2.2. Credit Assignment in MARL

Value decomposition methods by credit assignment (Sunehag et al., 2018; Rashid et al., 2018; Wang et al., 2021) are the most popular branches in the centralized training and decentralized execution (CTDE) (Oliehoek et al., 2008) paradigm. These methods should satisfy the individual-global-max (IGM) principle (Son et al., 2019) to guarantee the consistency between local and global greedy actions as

$$\arg\max_{\boldsymbol{u} \in \mathcal{U}^n} Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) = \begin{pmatrix} \arg\max_{u_1 \in \mathcal{U}} Q_1(\tau_1, u_1) \\ \vdots \\ \arg\max_{u_n \in \mathcal{U}} Q_n(\tau_n, u_n) \end{pmatrix}, \quad (1)$$

where $Q_{tot} \in \mathcal{Y}$ is the joint action value for each individual value function $Q_i(\tau_i, u_i)$. Under this principle, credit assignment aims to infer the contributions of predecessor value functions to Q_{tot} (Li et al., 2022a). The decomposition values $[Q_i]_{i=1}^n \in \mathcal{Q}$ are usually transformed into temporal values $[\widehat{Q}_k]_{k=1}^m$ via a human-designed function f_k with the global state $\boldsymbol{s}$, where m is the function number. It can represent a more general formulation as $Q_{tot} = \sum_{k=1}^m \alpha_k \widehat{Q}_k$ with the credit α_k, and we assume $[Q_i]_{i=1}^n$ and Q_{tot} are drawn following a kind of fixed (but unknown) distribution $\mathfrak{P} : \mathcal{Q} \to \mathcal{Y}$. The introduction of representative algorithms for the above formulation can be referred to in Appendix A.

3. Theoretical Analysis for Decomposition

Previous value-based studies have achieved great success in handling the joint action-value function to effectively enable CTDE in MARL. However, they often suffer from at least one of the following limitations: (1) Value decomposition ideas (Son et al., 2019; Wang et al., 2021; Iqbal et al., 2021; Rashid et al., 2020) with complex non-linear transformations may often fail to allow us to explicitly model the contribution of each agent or coalition of agents. (2) VDN (Sunehag et al., 2018), Qatten (Yang et al., 2020a), and SHAQ (Wang et al., 2022) measure the importance of each individual to the team, which ignores potentially all possible coalitions of all agents. (3) Existing mainstream value decomposition methods seldom diagnose whether individual agents can focus on specific information to help the mixer build a reasonable correlation of credit assignment between the global state $\boldsymbol{s}$ and the joint value function with a limited view of their surroundings. To resolve these problems, we propose a novel interpretable value decomposition method that uses the neural additive model to learn higher-order permutation relationships of each agent in terms of the local expansion of the joint action-value Q_{tot}, which achieves a better trade-off between performance and interpretability.

Following the general framework of the value decomposition method, we recall the joint action-value function and expand it in terms of Q_i by the Taylor expansion as

$$Q_{tot} = f_0 + \sum_{i=1}^{n} \alpha_i Q_i + \cdots + \sum_{i_1,\ldots,i_l}^{n} \alpha_{i_1\ldots i_l} \prod_{j=1}^{l} Q_{i_j} + \cdots, \tag{2}$$

where f_0 is a constant, all partial derivatives $\alpha_i = \frac{\partial Q_{tot}}{\partial Q_i}$ of order-1, and $\alpha_{i_1\ldots i_l} = \frac{1}{l!}\frac{\partial^l Q_{tot}}{\partial Q_{i_1}\ldots\partial Q_{i_l}}$ of order-l. In this term, it can be seen as a simple polynomial GAM expression (Dubey et al., 2022) with full n order interactions, which theoretically allows for learning any possible interaction order relationship among all agents. We enrich Eq. (2) with a general neural additive model (NAM) (Agarwal et al., 2021), as an extended GAM method, providing more precise predictions for the contribution of individual agents and coalitions of agents, which is formulated as

$$\begin{aligned} Q_{tot} = f_0 + \overbrace{\sum_{i=1}^{n} \alpha_i \underbrace{f_i(Q_i)}_{\text{order-1}}}^{\text{① similar to VDN}} + \cdots + \sum_{k\in\mathcal{D}_l} \alpha_k \underbrace{f_k(Q_k)}_{\text{order-}l} \\ + \cdots + \overbrace{\alpha_{1\ldots n} \underbrace{f_{1\ldots n}(Q_1,\ldots,Q_n)}_{\text{order-}n}}^{\text{② e.g., QMIX}}, \end{aligned} \tag{3}$$

where $f_k \in \{f_1, \cdots, f_{1\ldots n}\}^m$ is a shape function that transforms l local values Q_k into a temporal value $\widehat{Q}_k$, and $\mathcal{D}_l$ is the set of all non-empty subsets of $l \in \{1, \cdots, n\}$ with order-l interactions, i.e., $D_l = \{i_1 \ldots i_l\}$. When searching for a better value decomposition, we are often interested in this enrichment of the difference. To this end, we introduce the empirical risk minimizer $\widehat{Q}_{tot}$ in Eq (3) and the expected risk minimizer $Q^{\star}_{tot}$ in Eq (2) and consider $\mathcal{L}(\widehat{Q}_{tot}) - \mathcal{L}(Q^{\star}_{tot})$ as a regret bound. The conclusion shows that an upper bound always exists on our generalization according to regret analysis under the *1-Lipschitz* loss approximation. We provide approximation guarantees and detailed derivations for this type of enrichment, which can be found in Appendix B along with rigorous proofs.

Most existing MARL methods primarily focus on one of the terms in Eq. (3), aiming to maximize performance while neglecting the different orders of coalition among agents. For instance, VDN decomposes Q_{tot} into a sum of individual action values representing only a limited class with order-1, i.e., it is similar to term ① with equal credits. QMIX considers mixing all individual action values as the most effective value decomposition method falling under term ②, but it does not explicitly show its credit assignment. It is widely recognized (Lou et al., 2013; Chang et al., 2022) that it ceases to be interpretable with increasing order in Eq. (3), e.g., beyond pairwise interactions, albeit with some advantages to performance. Following this idea, we aim to maintain both the performance and interpretability of collaboration relationships in terms of any order of interaction by learning each unary and pairwise shape function as

$$Q_{tot} = f_0 + \sum_{i=1}^{n} \alpha_i f_i(Q_i) + \sum_{ij\in\mathcal{D}_2} \alpha_{ij} f_{ij}(Q_i, Q_j). \tag{4}$$

Furthermore, previous works (Rashid et al., 2018; Iqbal et al., 2021) typically learn credit α_k by the global state $\boldsymbol{s}$, with $P(Q_{tot}|\boldsymbol{s})$ calculated. It brings a spurious association between $\boldsymbol{s}$ and Q_{tot} that restricts deducing the contributions of individual agents and sub-teams from the overall success (Li et al., 2022a). One possible solution is to impose an intervention function on $\boldsymbol{s}$ by identifying the local history τ_i in an unobservable environment. Moreover, we are unsure whether the individual agent captures the important information that could help the mixer produce the credits from its observation instead of blindly pursuing performance. To this end, from the perspective of diagnosing the individual agent, we explicitly generate an individual semantic z_i from τ_i to achieve the identity representation, and then decentralized credit assignment is obtained by calculating

$$P(Q_{tot}|I(\boldsymbol{s})) = \sum_{\boldsymbol{z}} P(Q_{tot}|\boldsymbol{s},\boldsymbol{z})P(\boldsymbol{z}), \tag{5}$$

where $I(\cdot)$ denotes the intervention function and the joint semantic $\boldsymbol{z}$ is generated for backdoor adjustment by sampling individual semantics as $\boldsymbol{z} = [z_i \sim P(\tau_i)]_{i=1}^{n}$. It helps us diagnose whether individual agents focus on the relative

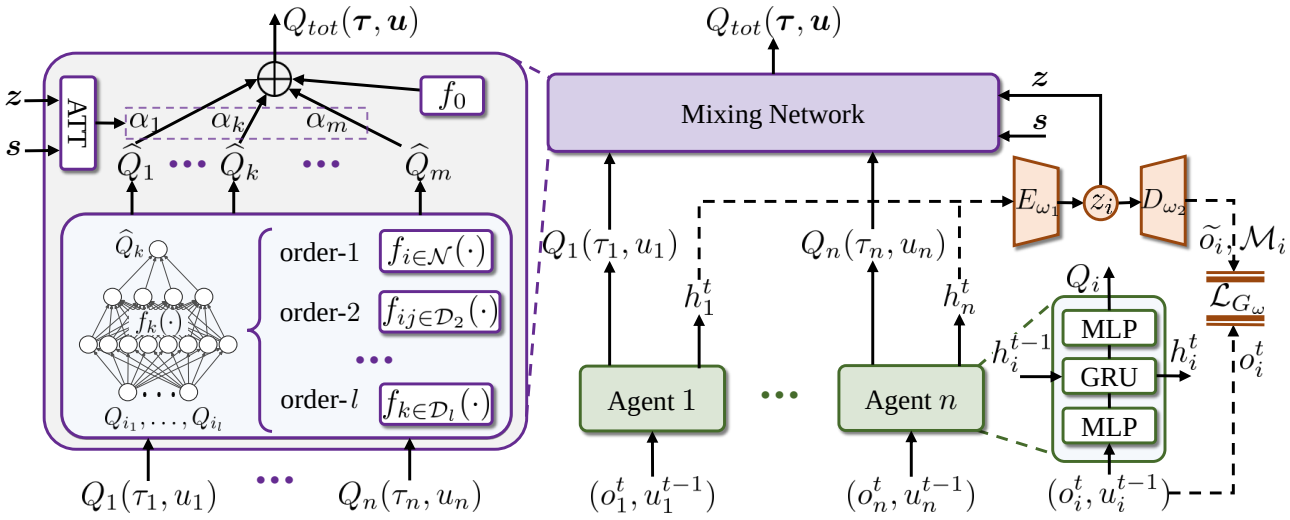


Figure 2. The overall framework of NA^{2}Q. First, each agent receives its local action-observation history τ_i and models its individual value function $Q_i(\tau_i, u_i)$. Next, we construct the identity semantic z_i by encoding τ_i, and take it together with the global state $\boldsymbol{s}$ to estimate credits, which provides a captured semantic interpretation. In the mixing network, we transform the local Q-values $[Q_i]_{i=1}^n$ into temporal Q-values $[\widehat{Q}_k]_{k=1}^m$ by the shape function f_k within order-l, where $l \in \mathcal{N}$, which are used to predict the joint Q-value together with credits.

importance of different task-relevant observations during the decision-making process in a more interpretable manner.

4. Neural Attention Additive Q-learning

Based on the previous analysis in Section 3, we propose a novel interpretable value decomposition method, called *Neural Attention Additive Q-learning* (NA^{2}Q), that explicitly learns a decomposition mapping for all possible order interactions among agents and captures semantic information from their observations. *Figure 2* illustrates the overall training procedure. For each agent, NA^{2}Q models a local value function $Q_i(\tau_i, u_i)$ and generates the identity semantic z_i by encoding the history τ_i. In this process, we maximize the resemblance of observations through decoding to ensure the accuracy of semantic information and upsample masks as an interpretation. In the mixer, the local Q-values $[Q_i]_{i=1}^n$ are transformed into $[\widehat{Q}_k]_{k=1}^m$ by all interactions of shape functions within order-l among agents, and the united semantics $\boldsymbol{z}$ and the global state $\boldsymbol{s}$ are fed into the intervention function to estimate credits. The joint value function is predicted depending on the temporal values $[\widehat{Q}_k]_{k=1}^m$ as well as credits. It can exactly model the contribution of any agent or coalition of agents to the overall success by enriching Eq. (2) with NAM.

Individual Action-Value Function. Following the mainstream works (Rashid et al., 2018; Wang et al., 2021), we employ a recurrent Q-function (Hausknecht & Stone, 2015) with parameter sharing for each agent i. Specifically, each function takes current local observation o_i^t with previous action u_i^{t-1} and previous hidden state h_i^{t-1} as inputs, and then outputs current hidden state h_i^t and local Q-value.

Constructing Identity Semantic. For each agent, we consider a general setting in which each agent focuses its observation on task-relevant regions. To capture this focus, we construct an underlying latent semantic from a local action observation of each agent via a variational auto-encoder (VAE) (Sohn et al., 2015), which can produce semantics corresponding to the importance assigned to each input observation. Over the course of training, the action-observation τ_i of each agent i is encoded by the VAE $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$ to sample its own identity semantic as $z_i = E_{\omega_1}(\tau_i)$. This semantic is then used as input for D_{ω_2} and upsampled to generate an attention mask as $\mathcal{M}_i = \varsigma(D_{\omega_2}(z_i))$, where $\varsigma(\cdot)$ represents the sigmoid function. Generally, the mask is interpreted to show where the agent is "looking" to make a decision (Shi et al., 2020). To maximize the resemblance between the identity semantic z_i and local observation o_i, the VAE G_ω is trained on a loss of the reconstruction observation along with a KL-divergence as

$$\mathcal{L}_{vae} = \sum_{i=1}^{n} \|o_i - \widetilde{o}_i\|_2^2 + D_{\mathrm{KL}}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(0, I)),$$

where $\widetilde{o}_i = \mathcal{M}_i \odot o_i$ and $\odot$ represent the overlaid observation with the mask and the element-wise multiplication, respectively. The normal distribution $\mathcal{N}(\mu, \sigma)$ is represented by deterministic functions, whose introduction is deferred to Appendix C. Meanwhile, the mask is expected to focus on as sparse and relevant region information as possible, so we apply a direct penalty to the mask by L_1-norm as

$$\mathcal{L}_{G_\omega} = \mathcal{L}_{vae} + \sum_{i=1}^{n} \|\mathcal{M}_i\|_1 . \tag{6}$$

By training the VAE with parameters $\omega = \{\omega_1, \omega_2\}$, we can obtain an attention mask to help humans better understand the local observation and latent identity semantic of each agent to influence the prediction of its action.

Learning Decomposition with Credit Assignment. To accomplish the decomposition formation in Eq. (4) and (5), we let Q_{tot} be decomposed into a neural GAM paradigm within order-2 by setting $l \leq 2$ as

$$Q_{tot} = f_0(\boldsymbol{s}) + \sum_{i=1}^{n} \alpha_i f_i(Q_i) + \sum_{i=1}^{n} \sum_{j>i}^{n} \alpha_{ij} f_{ij}(Q_i, Q_j), \tag{7}$$

where f_0 is a bias term, univariate and bivariate shape functions f_k are nonlinear functions (e.g., lightweight MLPs). To satisfy the IGM principle in Eq. (1), we restrict all the network weights to be non-negative by using the absolute in f_k. Considering that more efficient credit assignments can help local agents predict their actions more precisely, we also introduce the intervention function to realize decomposed training for backdoor adjustment. Specifically, the credit α_k is computed with the identity semantics $[z_i]_{i=1}^n$ and the global state $\boldsymbol{s}$ through a dot-product attention as

$$\alpha_k = [\alpha_i, \alpha_{ij}] = \frac{\exp((\boldsymbol{w}_z \boldsymbol{z})^\top \mathrm{ReLU}(\boldsymbol{w}_s \boldsymbol{s}))}{\sum_{k=1}^{m} \exp((\boldsymbol{w}_z \boldsymbol{z})^\top \mathrm{ReLU}(\boldsymbol{w}_s \boldsymbol{s}))}, \tag{8}$$

where $\boldsymbol{w}_s$, $\boldsymbol{w}_z$ are the learnable parameters, and ReLU is employed as the activation function. α_k is positive with softmax operation to ensure monotonicity.

Interpretability. Interpreting decomposition in Eq. (7) is intuitive as the influence of an individual Q-value on the prediction operates independently of other action values. It is possible to visualize the mapping relationships by visualizing the univariate shape function f_i, e.g., plotting Q_i on the x-axis and $\alpha_i f_i(Q_i)$ on the y-axis. The bivariate shape function f_{ij} is visualized through a heatmap (Lou et al., 2013; Radenovic et al., 2022), which is commonly used to achieve interpretation. Note that the visualization of the function accurately depicts how NA^{2}Q computes a prediction. In addition, the semantics of individual agents are upsampled into masks to represent feature importance, increasing the confidence of local observations on the semantics.

The overall learning objective is to end-to-end train the whole framework by minimizing the loss $\mathcal{L}$ with the mean squared temporal-difference (TD) error as

$$\mathcal{L}(\theta, \omega) = \|Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) - y\|_2^2 + \beta \mathcal{L}_{G_\omega} \tag{9}$$

where θ, ω are the whole framework parameters and β is a hyperparameter adjusting the weight of VAE loss. The target is estimated via Double DQN (Van Hasselt et al., 2016) as $y' = r + \gamma \overline{Q}_{tot}(\boldsymbol{\tau}', \arg\max_{\boldsymbol{u}' \in \mathcal{U}^n} Q_{tot}(\boldsymbol{\tau}', \boldsymbol{u}'))$. We summarize the pseudo-code of the proposed approach in Appendix D.

5. Experiments

In this section, we demonstrate our experimental results of NA^{2}Q on challenging tasks over LBF (Christianos et al., 2020) and SMAC (Samvelyan et al., 2019) benchmarks. The baselines that we select for comparison are nine popular value-based baselines, including VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2018), QTRAN (Son et al., 2019), Qatten (Yang et al., 2020a), QPLEX (Wang et al., 2021), Weighted QMIX (Rashid et al., 2020), DVD (Li et al., 2022a), CDS (Li et al., 2021a), and SHAQ (Wang et al., 2022). The implementation details of all algorithms are provided in Appendix F, along with the benchmarks. All graphs showing performance results for our method, baselines, and ablations study are plotted using mean $\pm$ std with five random seeds. Further, we present the interpretability of NA^{2}Q to render empirical evidence about which observations are of interest to the agents, as well as the contributions of each agent and coalition. The source code is available at `https://github.com/zichuan-liu/NA2Q`.

5.1. Level Based Foraging

We first run the experiments on two constructed LBF tasks, wherein agents navigate a 10×10 grid world and collect food by cooperating with other agents if needed. Each agent can observe a 5×5 sub-grid centering around it. When they cooperate to eat food that is smaller than their level at each step, they will receive a positive reward, otherwise, they will receive a negative reward of -0.002. The action space for each agent consists of movement in four directions, eating food, and a "none" action. We evaluate the performance of various algorithms with two quantities of agents and food.

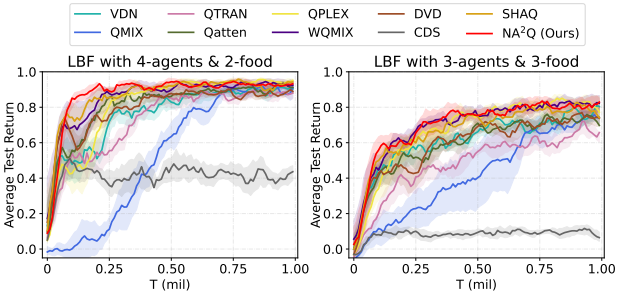


Figure 3. Average test return on two constructed tasks of LBF.

Performance on LBF. *Figure 3* shows the performance comparison against baselines on two constructed tasks of LBF. Our method achieves competitive performance in LBF tasks, demonstrating its efficiency across a range of scenarios. The failure of CDS may be due to the inability of diverse agents to explore collaborative strategies. VDN, QMIX, and QTRAN require more steps to discover sophisticated policies, indicating that they are in trouble due to the limitations of representing spurious relationships between

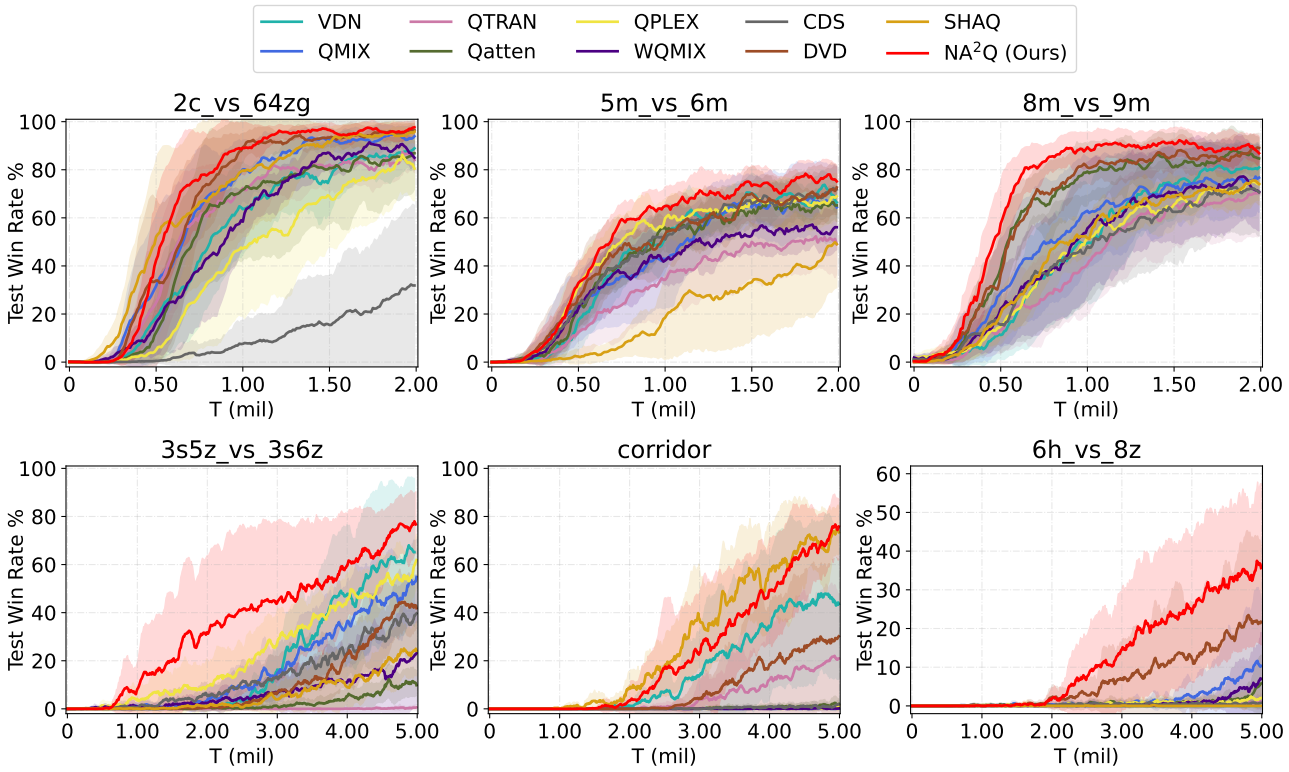


Figure 4. Test win rate % on hard (first row), and super hard (second row) maps of SMAC benchmark.

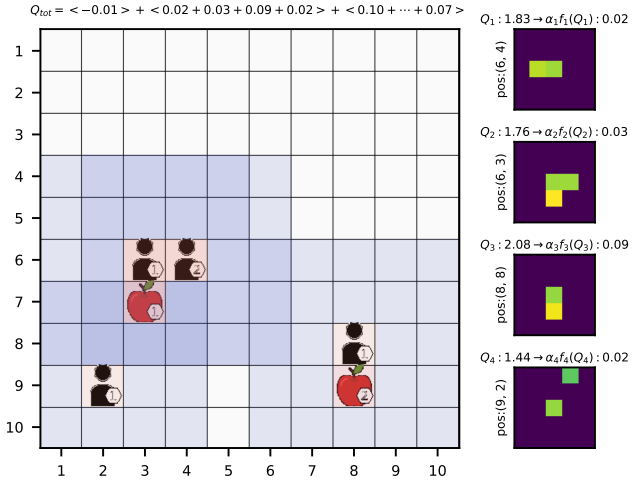
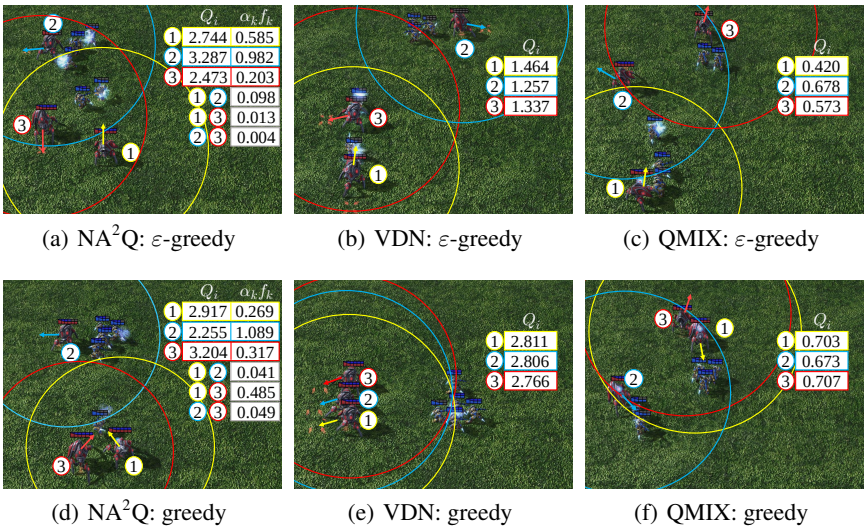


Figure 5. Visualization of the agent's mask at step 4, and the title indicates the corresponding credit assignment. The highlighted areas are the important regions for making decisions.

credits and decomposed Q-values. QPLEX receives a lower reward than NA^{2}Q before $0.5M$ timesteps, which may need more time steps to explore since the complex network architecture. Compared to QMIX, DVD obtains improved performance since it utilizes the de-confounded training mechanism. NA^{2}Q achieves slightly higher performance than Qatten. It implies that considering higher-order interactions and fine-grained learning semantics can promote credit assignment and correctly guide decentralized agents. Compared to WQMIX and SHAQ, NA^{2}Q achieves nearly the same performance with better robustness. The reason may be that providing a reasonable inference path for credit assignment can assist in improving coordination.

Interpretability of NA^{2}Q. To verify that NA^{2}Q possesses interpretability, we show its behavior matches that of corresponding agents on LBF. *Figure 5* illustrates the small regions that each agent focuses on, and the headings labeled to show its credits. It is evident that each agent captures task-relevant semantic information (the highlighted areas in the heatmaps) to make decisions. Specifically, agents pay more attention to the food position within their sight range, and tend to cooperate with teammates when the level of food is higher than themselves. Agents 1 and 2 only obtain credit assignments with 0.02 and 0.03, respectively, however, their pairwise shape function f_{12} obtains high credit with 0.10. This implies that they have captured the cooperative skill for eating the food, which should be attributed to considering the different orders of the coalition of agents in designing the value decomposition mechanism. Indeed, this semantic interpretation is also consistent with human visual patterns (Greydanus et al., 2018) that tend to focus selectively on parts of the visual space and form collaborative relationships. We likewise show the interpretation for



(a) NA^{2}Q: ε-greedy (b) VDN: ε-greedy (c) QMIX: ε-greedy

(d) NA^{2}Q: greedy (e) VDN: greedy (f) QMIX: greedy

Figure 6. Visualization of evaluation for NA^{2}Q and baselines on 3s_vs_5z map. Different colored circles indicate the corresponding central attack range, while arrows indicate movement or attack direction. Each decomposed Q-value is displayed at the top-right, and for NA^{2}Q, we report the contribution of its unary and pairwise shape functions to the team. This shows that the values of VDN and QMIX are difficult to explain, while the Q-values of the decomposition by NA^{2}Q intend to correspond more clearly to the actions.

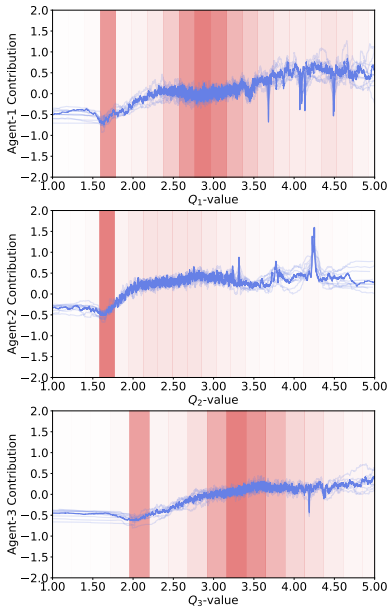


Figure 7. Learned shape function f_i by trained NA^{2}Q on 3s_vs_5z scenario. As expected, individual Q-values increase with the contribution of the agent.

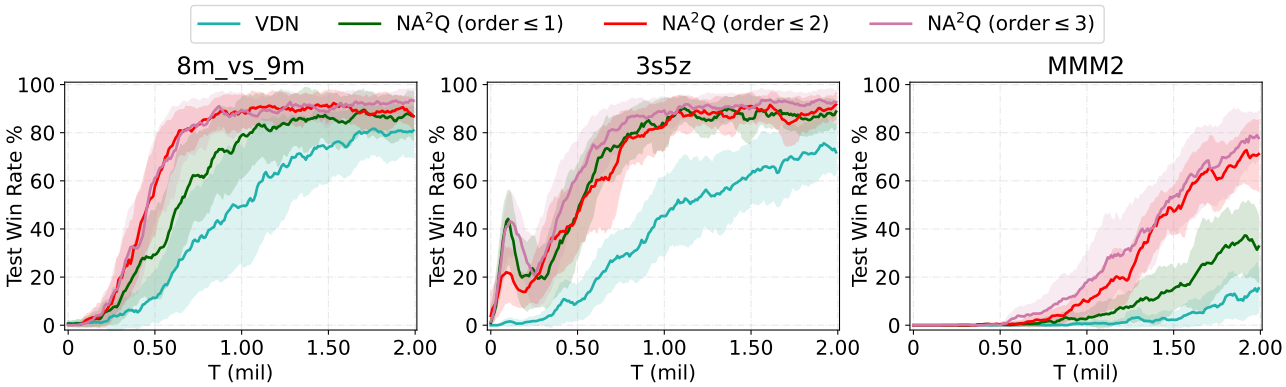
the whole episode, which can be found in Appendix G.

5.2. StarCraft Multi-Agent Challenge

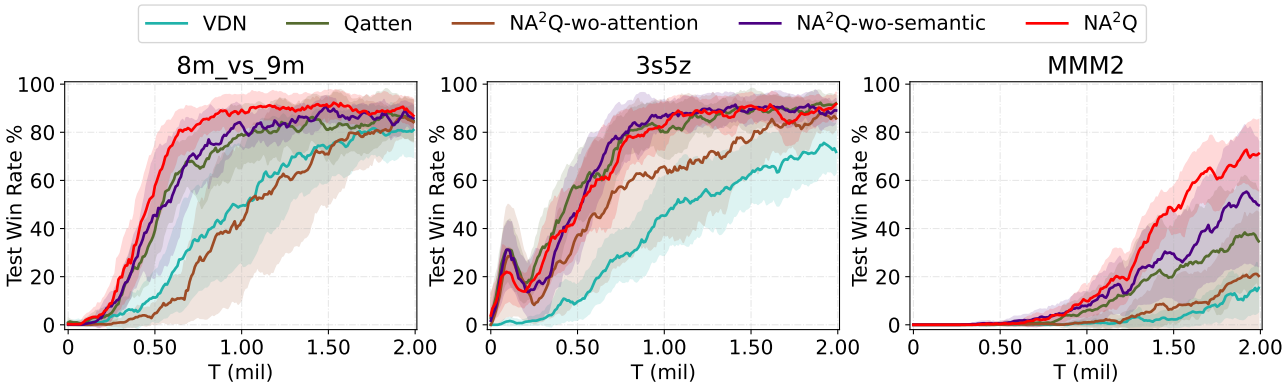
Further, to broadly compare the performance of NA^{2}Q with baselines, we conduct experiments on the more challenging SMAC benchmark, which is a commonly used testbed for MARL algorithms. At each timestamp, each agent receives local observations and then obtains a global reward after making a move or attacking its enemies. We compare the performance of NA^{2}Q with other baselines on 12 different scenarios, including easy, hard, and super hard scenarios. The details of these scenarios can be found in Appendix F.1.

Performance on SMAC. The experimental results for different scenarios are shown in *Figure 4*. We can find that NA^{2}Q could consistently gain almost the best performance on all scenarios, especially on the super hard tasks. QTRAN does not yield satisfactory performance, which may be due to the relaxation in practice that is insufficient for challenging domains. Both baseline VDN and QMIX can achieve satisfactory performance on some easy or hard maps, such as 5m_vs_6m, but in super hard maps they fail to well solve the tasks. Intuitively, super hard scenarios require more coordination skills, while their mixing network hardly captures the different interaction relationships among agents. Similarly, QPLEX and WQMIX do not perform well despite relaxed restrictions on the joint value function, which may contribute to inefficient value decomposition without considering the local semantics. Qatten falls short in satisfactory performance on super hard tasks, which implies that the lack of finely learned individual semantics brings about a spurious correlation between s and Q_{tot} and thus limits performance. One possible reason for CDS not performing as well as reported by Li et al. (2021a) is that paying more attention to policy diversity leads to instability during the learning stage, especially in less-agent maps, e.g., 2c_vs_64zg. SHAQ only achieves comparable performance with NA^{2}Q in the corridor map, which seems to have difficulty adapting to all scenarios. The reason could be that SHAQ ignores high-order interactions among agents. DVD only attains comparable performance with NA^{2}Q on 2c_vs_64zg map, and struggles to achieve competitive performance on the other scenarios, probably due to the fact that it neglects to explicitly consider high-order interactions among agents. In particular, for super hard task 6h_vs_8z, NA^{2}Q still maintains superior performance, while almost all the baselines are unable to learn efficient strategies. It validates that enriching shape functions for estimating credits over each agent and the coalition of agents can boost efficient value decomposition. In summary, our approach achieves impressive performance on all scenarios, showing the advantage of NA^{2}Q with attentive design. More empirical results can be referred to *Figure 12* in Appendix H.

Interpretability and Stability. To intuitively show the interpretability of NA^{2}Q, we display some keyframes on 3s_vs_5z scenario as shown in *Figure 6*. We first consider the suboptimal action by ε-greedy. As seen from *Figure 6(a)*, Agent-3 escapes from its teammates and receives a lower contribution relative to the allies, which can be understood as meaning that it does not contribute to the team. Mean-



(a) Number of interactive order terms



(b) Influence of identity semantics and attention

Figure 8. Ablation studies of NA^{2}Q on SMAC benchmark.

while, NA^{2}Q can also provide pairwise contributions among agents, whose contributions are close to 0 when they do not collaborate. However, it is hard to explicitly understand the behavior of VDN and QMIX from their Q-values. As for optimal actions of NA^{2}Q, Agent-2 plays an essential role in kiting enemies at this time step and obtains a higher contribution of 1.089, which is a crucial trick to victory that the agent can learn how to kite the enemies effectively (Samvelyan et al., 2019). An interesting finding is that Agent-1 and Agent-3 siege the enemy and their coalition contribution is remarkably higher than other coalitions. This shows the advantage of considering different orders of interactions among agents, which can facilitate deducing the contributions of each agent in value decomposition. Whereas VDN produces the same action and does not possess an explicit interpretation as NA^{2}Q since it only considers order-1 for credit assignment. For QMIX, the agents' behaviors are also difficult to understand because the Q-values are roughly equal. A detailed description of NA^{2}Q about observation semantics and agent contributions is deferred to Appendix I.

The interpretability of models is intrinsically coupled to their stability. To assess stability, we evaluate 10 models with different random seeds on 3s_vs_5z scenario, collecting 32 rounds of interaction data for each model and plotting the shape functions with semi-transparent lines. As shown in *Figure 7*, we display the average contribution of each shape function, where the blue lines and pink bars indicate the contribution of each agent to the team and the Q-value distribution, respectively, where a bar with an intense color means larger samples located here. As expected, the crimson area on the leftmost side represents its Q-value when an agent died, which means the agent had a lower contribution to the team. Most samples of Agent-1 and Agent-3 are gathered around 3.00 with larger positive contribution values, which implies that they spend more time step attacking enemies. We also find that most samples of Agent-2 are gathered with a negative contribution value. The reason may be that Agent-2 pays more attention to kiting the enemies, causing deaths to always occur earlier and having more dead Q-values. Moreover, we compute the standard deviation of the plotted shape functions to be 0.124, and the shape functions do not appear to deviate significantly, even for a few data points (white/light areas). This finding attests to the robustness and resilience of our novel value decomposition mechanism, further enhancing interpretability.

5.3. Ablation Study

To understand the impact of each component in the proposed method, we conduct ablation studies to answer the

following questions: (a) How does the model's performance benefit from the number of interaction orders among agents? (b) How do identity semantics influence performance? (c) Whether the intervention function is rational for value decomposition? To study component (a), we ablate shape functions for different order numbers named *NA^{2}Q (order$\leq l$)* in Eq. (3). Since higher-order interactions will decrease computational efficiency due to permutations, we select three different order interactions by setting $1 \leq l \leq 3$. To study components (b) and (c), *NA^{2}Q-w/o-semantic* represents replacing identity semantics $\boldsymbol{z}$ with global state $\boldsymbol{s}$ in Eq. (8), and *NA^{2}Q-w/o-attention* represents ablating attention mechanism for credit assignment, respectively. Additionally, since VDN and Qatten can be seen as special order-1 shape functions, we take them as a baseline for comparison.

We carry out ablation studies on three hard and super hard scenarios, and present the results in *Figure 8*. As shown in *Figure 8(a)*, NA^{2}Q achieves better performance as the number of order interactions increases, which validates the importance of considering higher-order interaction relationships among agents. Why not then have the number of order interactions as large as possible? A potential drawback is that an excessive number of order interactions might hurt interpretability, as shape functions beyond pairs are harder to visualize. Generally, moderate order terms (e.g., $l \leq 2$) are enough for an appropriate trade-off between performance improvement and interpretability. In *Figure 8(b)*, the ablation of each part of our intervention function brings a noticeable decrease in performance. Specifically, the performance of *NA^{2}Q-w/o-attention* and VDN decreases, which indicates that the global state information is beneficial to estimate the credit assignment. Besides, the performance of *NA^{2}Q-w/o-attention* is slightly higher than VDN because it considers more possible interactions among agents, leading to more capabilities than linear order-1 interactions. *NA^{2}Q-w/o-semantic* performs slightly worse than NA^{2}Q, which indicates the fine learning identity semantics own the greater representation ability to keep track of the feature influence of each agent. Additionally, the performance of *NA^{2}Q-w/o-semantic* is consistently superior to Qatten on a range of tasks, which implies that high-order interactions among agents can provide more capacity to search for efficient patterns of cooperation. To summarize, NA^{2}Q that is conditioned on all parts gives the best performance while retaining interpretability, which improves flexibility and saves human labor.

6. Conclusion

In this paper, we present NA^{2}Q in the scope of value decomposition, which combines the inherent interpretability of GAMs, opening the door for other advances in the interpretability perspective of MARL. NA^{2}Q allows for end-to-end training in a centralized fashion and models higher-order interactions to deduce precise credit for executing decentralized policies. Moreover, we provide local semantic masks as evidence for decision-making. The empirical results show that NA^{2}Q enjoys its interpretability and scalability while maintaining competitive performance. We believe that our work proves a solid basis for further research and could catalyze the community's effort toward understanding cooperative tasks. A promising direction for future work is improving the performance of NA^{2}Q by considering higher-order coalitions of agents. However, they might worsen the intelligibility of the learned NA^{2}Q with higher-order agent interactions, especially as the number of agents increases. It would be interesting to explore an efficient representation for interpreting a large-scale agent system, such as clustering similar terms in the NA^{2}Q framework.

7. Acknowledgements

The work was supported by the National Natural Science Foundation of China under Grant 62073160.

References

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. Neural additive models: Interpretable machine learning with neural nets. In *Advances in Neural Information Processing Systems*, pp. 4699–4711, 2021.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bastani, O., Pu, Y., and Solar-Lezama, A. Verifiable reinforcement learning via policy extraction. In *Advances in Neural Information Processing Systems*, pp. 2494–2504, 2018.

Bercu, B., Delyon, B., and Rio, E. *Concentration inequalities for sums and martingales*. Springer, 2015.

Berlinet, A. and Thomas-Agnan, C. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

Blumenkamp, J. and Prorok, A. The emergence of adversarial communication in multi-agent reinforcement learning. In *Proceedings of the Conference on Robot Learning*, pp. 1394–1414, 2021.

Chang, C.-H., Caruana, R., and Goldenberg, A. Node-GAM: Neural generalized additive model for interpretable deep learning. In *Proceedings of the International Conference on Learning Representations*, pp. 1–25, 2022.

Chen, L., Zhang, H., Xiao, J., He, X., Pu, S., and Chang, S.-F. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4613–4623, 2019.

Christianos, F., Schäfer, L., and Albrecht, S. Shared experience actor-critic for multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 10707–10717, 2020.

Dubey, A., Radenovic, F., and Mahajan, D. Scalable interpretability via polynomials. In *Advances in Neural Information Processing Systems*, pp. 36748–36761, 2022.

Ghorbani, A., Abid, A., and Zou, J. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3681–3688, 2019.

Glymour, M., Pearl, J., and Jewell, N. P. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Greydanus, S., Koul, A., Dodge, J., and Fern, A. Visualizing and understanding atari agents. In *Proceedings of the International Conference on Machine Learning*, pp. 1792–1801, 2018.

Hastie, T. and Tibshirani, R. Generalized additive models. *Statistical Science*, 1(3):297–318, 1986.

Hausknecht, M. and Stone, P. Deep recurrent Q-learning for partially observable MDPs. In *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents*, pp. 29–37, 2015.

Iqbal, S., De Witt, C. A. S., Peng, B., Böhmer, W., Whiteson, S., and Sha, F. Randomized entity-wise factorization for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 4596–4606, 2021.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23 (6):4909–4926, 2021.

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Li, C., Wang, T., Wu, C., Zhao, Q., Yang, J., and Zhang, C. Celebrating diversity in shared multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3991–4002, 2021a.

Li, J., Kuang, K., Wang, B., Liu, F., Chen, L., Wu, F., and Xiao, J. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 934–942, 2021b.

Li, J., Kuang, K., Wang, B., Liu, F., Chen, L., Fan, C., Wu, F., and Xiao, J. Deconfounded value decomposition for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 12843–12856, 2022a.

Li, P., Tang, H., Yang, T., Hao, X., Sang, T., Zheng, Y., Hao, J., Taylor, M. E., Tao, W., and Wang, Z. PMIC: improving multi-agent reinforcement learning with progressive mutual information collaboration. In *Proceedings of the International Conference on Machine Learning*, pp. 12979–12997, 2022b.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, pp. 1–14, 2016.

Liu, Z., Zhu, Y., Wang, Z., Gao, Y., and Chen, C. MIXRTs: Toward interpretable multi-agent reinforcement learning via mixing recurrent soft decision trees. *arXiv preprint arXiv:2209.07225*, 2022.

Lou, Y., Caruana, R., Gehrke, J., and Hooker, G. Accurate intelligible models with pairwise interactions. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 623–631, 2013.

Massart, P. Some applications of concentration inequalities to statistics. In *Proceedings of the Annales de la Faculté des sciences de Toulouse: Mathématiques*, pp. 245–303, 2000.

Oliehoek, F. A. and Amato, C. *A concise introduction to decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer, 2016.

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

Radenovic, F., Dubey, A., and Mahajan, D. Neural basis models for interpretability. In *Advances in Neural Information Processing Systems*, pp. 8414–8426, 2022.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 4295–4304, 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted QMIX: Expanding monotonic value function

factorisation for deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 10199–10210, 2020.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning*, pp. 1278–1286, 2014.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. The starcraft multi-agent challenge. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2186–2188, 2019.

Shi, W., Huang, G., Song, S., Wang, Z., Lin, T., and Wu, C. Self-supervised discovering of interpretable features for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2712–2724, 2020.

Silva, A., Gombolay, M., Killian, T., Jimenez, I., and Son, S.-H. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 1855–1865, 2020.

Slack, D., Hilgard, A., Singh, S., and Lakkaraju, H. Reliable post hoc explanations: Modeling uncertainty in explainability. In *Advances in Neural Information Processing Systems*, pp. 9391–9404, 2021.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pp. 3483–3491, 2015.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 5887–5896, 2019.

Sun, X., Wang, Z., Ding, R., Han, S., and Zhang, D. pureGAM: Learning an inherently pure additive model. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1728–1738, 2022.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018.

Tsang, M., Liu, H., Purushotham, S., Murali, P., and Liu, Y. Neural interaction transparency (NIT): Disentangling learned interactions for improved interpretability. In *Advances in Neural Information Processing Systems*, pp. 5804–5813, 2018.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2094–2100, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

Wang, J., Zhang, Y., Kim, T.-K., and Gu, Y. Shapley Q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7285–7292, 2020.

Wang, J., Ren, Z., Liu, T., Yu, Y., and Zhang, C. QPLEX: Duplex dueling multi-agent Q-learning. In *Proceedings of the International Conference on Learning Representations*, pp. 1–27, 2021.

Wang, J., Wang, J., Zhang, Y., Gu, Y., and Kim, T.-K. SHAQ: Incorporating shapley value theory into multi-agent Q-learning. In *Advances in Neural Information Processing Systems*, pp. 5941–5954, 2022.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. Mean field multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 5571–5580, 2018.

Yang, Y., Hao, J., Liao, B., Shao, K., Chen, G., Liu, W., and Tang, H. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020a.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. I. On function approximation in reinforcement learning: Optimism in the face of large state spaces. In *Advances in Neural Information Processing Systems*, pp. 13903–13916, 2020b.

Zahavy, T., Ben-Zrihem, N., and Mannor, S. Graying the black box: Understanding DQNs. In *Proceedings of the International Conference on Machine Learning*, pp. 1899–1908, 2016.

A. Credit Assignment for Value Decomposition Algorithms

Previous work (Li et al., 2022a) defined the general formula for credit assignment in value decomposition methods as

$$Q_{tot} = \sum_{k=1}^{m} \alpha_k \widehat{Q}_k, \tag{10}$$

where $\widehat{Q}_k$ is transformed as a temporal value by $f_k(\cdot)$ and α_k denotes a credit that expresses the contribution of the temporal value to the joint action-value Q_{tot}. This formula can be applied for generalization in widely investigated approaches of mixing networks based on value decomposition, next we introduce these methods in detail[1].

The first method is VDN (Sunehag et al., 2018), which seeks to learn a joint value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$ via equal credit assignment. It represents Q_{tot} as the sum of all individual value functions as $Q_{tot} = \sum_{i=1}^{n} Q_i$ without the use of additional state information., where Eq. (10) can be rewritten when $m = n$, $\alpha_k = 1$, and $\widehat{Q}_i = f_i(Q_i)$.

More common algorithms transform the local Q-values into the temporal Q-values via the global state $\boldsymbol{s}$. For example, QMIX (Rashid et al., 2018) can be represented by a monotonic neural network $f_{1\ldots n}(\cdot)$ with the global state $\boldsymbol{s}$ as

$$[\widehat{Q}_k]_{k=1}^{m} = f_{1\ldots n}(Q_1, \ldots, Q_n), \frac{\partial f_{1\ldots n}}{\partial Q_i} > 0,$$

where $k \in \{1, \cdots, m\}$ denotes the embedding number. Then the credit $\alpha_k(\boldsymbol{s})$ is calculated by another monotonic neural network and utilized in Eq. (10). Some methods that improve on QMIX, e.g, Qatten (Yang et al., 2020a) replace the neural network $f_{1\ldots n}$ into an attention mechanism, Weighted QMIX (Rashid et al., 2020) uses different weights on TD error, and CDS (Li et al., 2021a) improves diversity among agents by constructing intrinsic rewards.

Further, QPLEX (Wang et al., 2021) combines QMIX and VDN in a dueling mixing network as

$$Q_{tot} = \sum_{i}^{n} \alpha_i \widehat{Q}_i + \alpha_{1\ldots n} f_{1\ldots n}(Q_1, \ldots, Q_n), \frac{\partial f_{1\ldots n}}{\partial Q_i} > 0,$$

where $\widehat{Q}_i = Q_i$ represents the local temporal value and $f_{1\ldots n}$ represents the advantage function to get $\widehat{Q}_{1\ldots n}$, which also uses an attention mechanism. Therefore, it is equivalent to Eq. (10) when $m = n + 1$ and $\alpha_k \in \{\alpha_1, \cdots, \alpha_n, \alpha_{1\ldots n}\}^m$. It is straightforward to notice that QPLEX is the sum of term ① and term ② in Eq. (3).

The last method SHAQ (Wang et al., 2022) improves the credit assignment of QMIX via Shapley theory for interpretation, which can also be expressed by Eq. (10).

B. Approximation Guarantees for NA^{2}Q

Inspired by non-linear GAMs, e.g., NAM (Agarwal et al., 2021) and SPAM (Dubey et al., 2022), we modify the decomposition of $Q = [Q_i]_{i=1}^{n} \in \mathcal{Q}$ in Eq. 3 by rewriting the order number $1 \leq l \leq n$ with the shape functions as

$$Q_{tot} = f_0 + \lambda_{1d} \cdot \langle \boldsymbol{a}_{1d}, F_1(Q) \rangle + \sum_{d=1}^{\rho_2} \lambda_{2d} \cdot \langle \boldsymbol{a}_{2d}, F_2(Q) \rangle^2 + \cdots + \sum_{d=1}^{\rho_n} \lambda_{nd} \cdot \langle \boldsymbol{a}_{nd}, F_n(Q) \rangle^n, \tag{11}$$

where $\{\lambda_{ld}\}_{d=1}^{\rho_l}$ and $\{\boldsymbol{a}_{ld}\}_{d=1}^{\rho_l}$ are the corresponding eigenvalues and bases for credit matrix $\boldsymbol{\alpha}_l = \{\alpha_{\mathcal{D}_l}\}$ to represent the order-l interactions between all non-empty subsets of $l \in \mathcal{N}$, $\rho_l \in \{1, \rho_2, \cdots, \rho_n\}$ denotes the rank of the tensor, and the function $F_l(Q) = [f_{l1}(\cdot), f_{l2}(\cdot), \cdots, f_{ln}(\cdot)] \in \mathcal{F}_l$ is a family of shape functions in the order-l. Next, we present learning-theoretic and approximation guarantees for this type of enrichment, with a more precise regret bound.

Assumption B.1. (η - Exponential Spectral Decay of Approximation.) For the family of all decomposition $Q \in \mathcal{Q}$ as outlined in Eq. (11), we assume that there exist absolute constants $C_1 < 1$ and $C_2 = \mathcal{O}(1)$ such that $\lambda_{ld} \leq C_1 \exp(-C_2 \cdot d^{\eta})$ for each $l \in \mathcal{N}$ and $d \geq 1$.

Assumption B.1 provides a soft threshold for singular value decay, i.e., implying that only a few decay degrees of freedom are sufficient to accurately approximate f_k. We consider the general results under the *1-Lipschitz* loss approximated by

[1]For convenience, all bias networks are omitted if existing.

this enrichment decomposition of the metric regret bound. Let us denote the Taylor expansion decomposition in Eq. (2) as $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) : \mathcal{Q} \to \overline{\mathcal{Y}}$. Thus, we aim to bound the expected risk in Eq. (2) with the empirical risk in Eq. (3) to demonstrate that learning an enrichment decomposition method does not incur a larger error compared with learning the Taylor expansion. At a high level, for any function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) : \mathcal{Q} \to \mathcal{Y}$ and bounded *1-Lipschitz* loss $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$, the empirical risk over b samples from $\mathcal{B}$ as $\widehat{\mathcal{L}}_b(Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})) = \frac{1}{b}\sum_{j=1}^{b} \ell(Q_{tot}, y)$. We donate $\widehat{Q}_{tot}$ as the *empirical risk minimizer*, then,

$$\widehat{Q}_{tot} = \arg\min_{Q_{tot} \in \mathcal{Y}} \widehat{\mathcal{L}}_b(Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})). \tag{12}$$

Similarly, the expected risk can be given, over the sample distribution $\mathfrak{P}$ as $\mathcal{L}(Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})) = \mathbb{E}_{([Q_i]_{i=1}^n, y) \sim \mathfrak{P}}[\ell(Q_{tot}, y)]$. Then we have that the optimal *expected risk minimizer* $Q^\star_{tot}$ as

$$Q^\star_{tot} = \arg\min_{Q_{tot} \in \overline{\mathcal{Y}}} \mathcal{L}(Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})). \tag{13}$$

Our preparation is complete, so we can now discuss the regret bound for our generalization. We state the full Theorem here.

Theorem B.2. *Let ℓ be 1-Lipschitz, $\delta \in (0, 1]$ and Assumption B.1 hold with constants $\{C_1, C_2, \eta\}$. Then, for L_1-norm models, where $\|\boldsymbol{a}_{ld}\|_1 \leq B_a, 1 \leq l \leq n$, and $\|\boldsymbol{\lambda}\|_1 \leq B_\lambda$ where $\boldsymbol{\lambda} = \{\{\lambda_{ld}\}_{d=1}^{\rho_l}\}_{l=1}^n$, there exists some absolute constants $\{C_1, C_2\}$ with probability at least $1 - \delta, \delta \in (0, 1]$ that we have*

$$\mathcal{L}(\widehat{Q}_{tot}) - \mathcal{L}(Q^\star_{tot}) \leq 2B_\lambda \cdot \left(\sum_{l=1}^{n} (B_a)^l\right) \sqrt{\frac{\log(n)}{b}} + \frac{C_1}{C_2} \cdot \left(\sum_{l=1}^{n} \exp(-\rho_l^\eta)\right) + 2(\sqrt{2}+1) \cdot \sqrt{\frac{\log(2/\delta)}{b}}. \tag{14}$$

Proof. For the expected function $Q^\star_{tot}$, we also denote the corresponding eigenvalues as $\{\{\lambda^\star_{ld}\}_{d=1}^{\overline{\rho}_l}\}_{l=1}^n$ and bases as $\{\{\boldsymbol{a}^\star_{ld}\}_{d=1}^{\overline{\rho}_l}\}_{l=1}^n$. Consider the $\widetilde{Q}_{tot} \in \mathcal{Y}$ that is a "truncated" version of the optimal $Q^\star_{tot}$. Therefore, we can rewrite the regret bound as

$$\mathcal{L}(\widehat{Q}_{tot}) - \mathcal{L}(Q^\star_{tot}) = \underbrace{\mathcal{L}(\widehat{Q}_{tot}) - \widehat{\mathcal{L}}_b(\widehat{Q}_{tot})}_{①} + \underbrace{\widehat{\mathcal{L}}_b(\widehat{Q}_{tot}) - \widehat{\mathcal{L}}_b(\widetilde{Q}_{tot})}_{\leq 0} + \underbrace{\widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) - \mathcal{L}(Q^\star_{tot})}_{②},$$

where the middle term $\widehat{\mathcal{L}}_b(\widehat{Q}_{tot}) - \widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) \leq 0$ since $\widehat{Q}_{tot}$ minimizes the empirical risk in Eq. (13). Therefore, binding on terms ① and ② can provide us with a proof of the bound. The bound for term ② is tractable, which can be proved via Lemma B.3. Hence with probability at least $1 - \delta, \delta \in (0, 1]$, we have that

$$\widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) - \mathcal{L}(Q^\star_{tot}) \leq \sum_{l=1}^{n} \frac{C_1}{C_2} \cdot \exp(-\rho_l^\eta) + 2\sqrt{\frac{\log(2/\delta)}{b}}. \tag{15}$$

Then inspired by Radenovic et al. (2022), we handle the term ① via bounding the Rademacher complexity (Wainwright, 2019). The loss function ℓ is Lipschitz and bounded, with probability at least $1 - \delta$ for any $\delta \in (0, 1]$ over samples of length b. These conditions allow us to apply Theorem 8 and Theorem 12 from Bartlett & Mendelson (2002), whose proof uses McDiarmid's inequality. Thus we have that

$$\mathcal{L}(\widehat{Q}_{tot}) - \widehat{\mathcal{L}}_b(\widehat{Q}_{tot}) \leq \mathcal{R}_b(\ell \circ \mathcal{F}) + \sqrt{\frac{8\log(2/\delta)}{b}},$$

where $\mathcal{F}$ denotes the set of all joint value functions represented, i.e, $\forall Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) \in \mathcal{F}$, and $\mathcal{R}_b$ is the empirical Rademacher complexity. According to the Theorem 12 from Bartlett & Mendelson (2002), $\mathcal{R}_b(\ell \circ \mathcal{F}) \leq 2L \cdot \mathcal{R}_b(\mathcal{F}) \leq 2L \cdot \sum_{l=1}^{n} \mathcal{R}_b(\mathcal{F}_l)$. Thus, we can put all the order terms together since ℓ is L-Lipschitz, and rewrite the above equation as

$$\mathcal{L}(\widehat{Q}_{tot}) - \widehat{\mathcal{L}}_b(\widehat{Q}_{tot}) \leq 2L \cdot \sum_{i=1}^{n} \mathcal{R}_b(\mathcal{F}_i) + 2\sqrt{2} \cdot \sqrt{\frac{\log(2/\delta)}{b}},$$

where $\mathcal{F}_l$ denotes the family of $F_l(\cdot)$ in the order-l. Therefore, since we consider the L_1-norm models, there exist eigenvalue $\|\boldsymbol{\lambda}\|_1 \leq B_\lambda$ and base vector $\|\boldsymbol{a}_{ld}\|_1 \leq B_a$, where $\forall l \in \mathcal{N}$ and $\forall d \in \{1, \cdots, \rho_l\}$. Under these constraints, the term ① can bound the empirical Rademacher complexity via Lemma 3 from Dubey et al. (2022) and Lemma 5.2 from Massart (2000), and we have

$$\mathcal{L}(\widehat{Q}_{tot}) - \widehat{\mathcal{L}}_b(\widehat{Q}_{tot}) \leq 2B_\lambda \cdot \left(\sum_{l=1}^{n} (B_a)^l\right) \cdot \sqrt{\frac{\log(n)}{b}} + 2\sqrt{2} \cdot \sqrt{\frac{\log(2/\delta)}{b}}. \tag{16}$$

Finally, the bound for combining Eq. (15) and Eq. (16) provides us with the results of the proof. □

Lemma B.3. *With probability at least $1-\delta$ for any $\delta \in (0,1]$ and some absolute constants $\{C_1, C_2\}$, we have that*

$$\widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) - \mathcal{L}(Q^\star_{tot}) \leq \sum_{l=1}^{n} \frac{C_1}{C_2} \cdot \exp(-\rho_l^\eta) + 2\sqrt{\frac{\log(2/\delta)}{b}}.$$

Proof. Observe,

$$\begin{aligned}\widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) - \mathcal{L}(Q^\star_{tot}) =& \widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) - \mathcal{L}(\widetilde{Q}_{tot}) + \mathcal{L}(\widetilde{Q}_{tot}) - \mathcal{L}(Q^\star_{tot}) \\ \leq& \underbrace{\left|\widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) - \mathcal{L}(\widetilde{Q}_{tot})\right|}_{\text{(2a)}} + \underbrace{\left|\mathcal{L}(\widetilde{Q}_{tot}) - \mathcal{L}(Q^\star_{tot})\right|}_{\text{(2b)}}.\end{aligned}$$

To bound (2a), we have sample points $\in \mathfrak{P}$ in a batch b that satisfies $\mathcal{L}(\widetilde{Q}_{tot}) = \mathbb{E}[\ell(\widetilde{Q}_{tot}, y)]$, where $0 \leq \ell(\cdot,\cdot) \leq 1$. Hence we employ Azuma-Hoeffding's inequality (Bercu et al., 2015) and substitute the reproducing Hilbert space (RHS) (Berlinet & Thomas-Agnan, 2011) probability with $1-\delta$, which can be rewritten in terms as

$$\left|\widehat{\mathcal{L}}_b(\widetilde{Q}_{tot}) - \mathcal{L}(\widetilde{Q}_{tot})\right| \leq 2\sqrt{\frac{\log(2/\delta)}{b}}.$$

Since ℓ is L-Lipschitz, we have for some $\{\widetilde{Q}_{tot}, Q^\star_{tot}, y\} \in \mathcal{Y}$,

$$\begin{aligned}\left|\ell(\widetilde{Q}_{tot}, y) - \ell(Q^\star_{tot}, y)\right| \leq& \left|L \cdot |\widetilde{Q}_{tot} - y| - L \cdot |Q^\star_{tot} - y|\right| \\ =& L \cdot \left||\widetilde{Q}_{tot} - y| - |Q^\star_{tot} - y|\right| \\ \leq& L \cdot \left|\widetilde{Q}_{tot} - Q^\star_{tot}\right|.\end{aligned}$$

Thus, when $L = 1$, the bound (2b) is derived as

$$\begin{aligned}\left|\mathcal{L}(\widetilde{Q}_{tot}) - \mathcal{L}(Q^\star_{tot})\right| \leq& \left|\mathbb{E}_{([Q_i]_{i=1}^n, y)\sim\mathfrak{P}}[\ell(\widetilde{Q}_{tot}, y) - \ell(Q^\star_{tot}, y)]\right| \\ \leq& \mathbb{E}_{([Q_i]_{i=1}^n, y)\sim\mathfrak{P}}\left[|\ell(\widetilde{Q}_{tot}, y) - \ell(Q^\star_{tot}, y)|\right] \\ \leq& L \cdot \mathbb{E}_{([Q_i]_{i=1}^n, y)\sim\mathfrak{P}}\left[|\widetilde{Q}_{tot} - Q^\star_{tot}|\right] \\ \leq& L \cdot \sup_{Q\in\mathcal{Q}} |\widetilde{Q}_{tot} - Q^\star_{tot}| \\ =& \sup_{Q\in\mathcal{Q}} |\widetilde{Q}_{tot} - Q^\star_{tot}|.\end{aligned}$$

Observing now that $\forall Q \in \mathcal{Q}$, we have

$$\begin{aligned}\left|\widetilde{Q}_{tot} - Q^\star_{tot}\right| =& \left|\sum_{l=1}^{n}\sum_{d=\rho_l}^{\overline{\rho}_l} \lambda^\star_{ld} \cdot \langle \boldsymbol{a}^\star_{ld}, F_l(Q)\rangle^l\right| \\ \leq& \sum_{l=1}^{n}\sum_{d=\rho_l}^{\overline{\rho}_l} \left|\lambda^\star_{ld} \cdot \langle \boldsymbol{a}^\star_{ld}, F_l(Q)\rangle^l\right| \\ \leq& \sum_{l=1}^{n}\sum_{d=\rho_l}^{\overline{\rho}_l} |\lambda^\star_{ld}|,\end{aligned}$$

when hold on Assumption B.1, we have that $\lambda_{ld} = C_1 \exp(-C_2 \cdot d^\eta)$ if obeys the η-exponential spectral decay. Thus,

$$\sum_{l=1}^{n}\sum_{d=\rho_l}^{\overline{\rho}_l} |\lambda^\star_{ld}| \leq \sum_{l=1}^{n}\sum_{d=\rho_l}^{\overline{\rho}_l} C_1 \exp(-C_2 \cdot d^\eta) \leq \sum_{l=1}^{n}\int_{d=\rho_l}^{\infty} C_1 \exp(-C_2 \cdot d^\eta).$$

Since $\eta \geq 1$, we can bound by the Eq. (E.16) from Yang et al. (2020b) with the RHS as

$$\left|\widetilde{Q}_{tot} - Q^{\star}_{tot}\right| \leq \sum_{l=1}^{n} \int_{d=\rho_l}^{\infty} C_1 \exp(-C_2 \cdot d^{\eta}) \leq \sum_{l=1}^{n} \frac{C_1}{C_2} \exp(-\rho_l^{\eta}).$$

Therefore, we finish the proof of Lemma B.3.

□

C. Variational Auto-Encoder Background

A variational auto-encoder (VAE) (Sohn et al., 2015) is a popular generative model to learn an attention mask, e.g., U-Net (Ronneberger et al., 2015) for semantic segmentation. VAE aims to maximize the marginal log-likelihood $\log p(T) = \sum_{j=1}^{b} \log p(\tau^j)$, where $T = [\tau^j]_{j=1}^{b} \in \mathcal{T}$ denotes the set of local action-observation histories from $\mathcal{B}$, and it is common to replace the optimized variational lower-bound as

$$\log p(T) \geq \mathbb{E}_{q(T|z)} \left[\log p(T|z)\right] + D_{\mathrm{KL}}(q(z|T)||p(z)),$$

where $p(z)$ generally is a multivariate normal distribution $\mathcal{N}(0, I)$ to represent the prior. We define the posterior $q(z|T) = \mathcal{N}(z|\mu, \sigma^2(T)I)$ as the encoder E_{ω_1} and $p(T|z)$ as the decoder D_{ω_2}. It is understood that given a sample τ is fed into the VAE to produce a latent semantic vector z, and then this vector is reconstructed into the desired sample by training. To apply gradient descent on the variational lower-bound, we allow the re-parametrization trick (Rezende et al., 2014) to train on a reconstruction loss with a KL-divergence as

$$\mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma)} \left[f(z)\right] = \mathbb{E}_{\nu\sim\mathcal{N}(0,I)} \left[f(\mu + \sigma\nu)\right].$$

Thus μ and σ can be represented by deterministic functions, allowing for back-propagation.

D. Pseudo Code

Algorithm 1 Neural Attention Additive Q-learning

Initialize a set of agents $\mathcal{N} = \{1, 2, \cdots, n\}$
Initialize networks of local agents $Q_i(\tau_i, u_i; \theta)$ and target networks $Q_i(\tau_i', u_i'; \hat{\theta})$, $G_{\hat{\omega}}$ with $\hat{\theta} \leftarrow \theta$
Initialize a VAE $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$ with parameters ω
Initialize a replay buffer $\mathcal{B}$ for storing episodes
repeat
  Initialize a history embedding h_i^0 and an action vector u_i^0 for each agent
  Observe each agent's partial observation $\left[o_i^1\right]_{i=1}^{n}$
  for $t = 1 : T$ **do**
    Get $\tau_i^t = \left\{o_i^t, h_i^{t-1}\right\}$ for each agent and calculate the individual value function $Q_i(\tau_i^t, u_i^{t-1})$
    Get the hidden state h_i^t and select action u_i^t via value function with probability ε exploration
    Unsampled n identity semantic masks $[\mathcal{M}_i \sim G_\omega(h_i^t)]_{i=1}^{n}$ as an interpretation
    Execute u_i^t to receive the reward r^t, next state $\boldsymbol{s}^{t+1}$
  end for
  Store the episode trajectory to $\mathcal{B}$
  Sample a batch of episodes trajectories with batch size b from $\mathcal{B}$
  for $t = 1 : T$ **do**
    Calculate $\mu, \sigma = E_{\omega_1}(\tau_i^t)$ and identity semantics $\boldsymbol{z} = [z_i \sim \mathcal{N}(\mu, \sigma)]_{i=1}^{n}$
    Get $\widetilde{o_i} = \mathcal{M}_i \odot o_i$ and calculate $\mathcal{L}_{G_\omega}$ via Eq. (6)
    Get the attention weight $\alpha_k(\boldsymbol{z}, \boldsymbol{s})$ by the intervention function in Eq. (8)
    Calculate the joint value function within order-2 interactions via Eq. (7)
  end for
  Construct the loss function defined in Eq. (9)
  Update ω and θ by minimizing the above loss
  Periodically update $\hat{\theta} \leftarrow \theta$
until $Q_i(\tau_i, u_i; \theta)$ converges

E. Related work

Value Decomposition in MARL. Since the joint action space grows exponentially in proportion to the number of participating agents (Yang et al., 2018), the centralized training and decentralized execution (CTDE) (Oliehoek et al., 2008) paradigm is proposed to relieve this issue and become a mainstream framework in MARL. One of the crucial challenges in CTDE is credit assignment, which aims to infer how much each agent contributes to the overall success. Under the CTDE framework, VDN (Sunehag et al., 2018) assumes that any joint action-value function can be decomposed into a linear summation of individual value functions. Nevertheless, this equivalent factorization limits the credit assignment of the global Q-value. To mitigate this issue, some implicit credit assignment methods, e.g., QMIX (Son et al., 2019) and QTRAN (Wang et al., 2021), represent the joint value function into a richer family for value decomposition with complex nonlinear transformation function. Further, Weighted QMIX (Rashid et al., 2020) proposes a weighted projection to decompose the joint action-value function, and PMIC (Li et al., 2022b) utilizes more effective mutual information to collaborate better. However, these methods neglect causal explanations in credit assignment, which may be unreasonable since suboptimal actions lack an explicit reasoning mechanism. They entangle the interactions at temporal hidden layers for credit assignment. Thus, recent works (Wang et al., 2022; Li et al., 2021b) apply the Shapley theory to trustworthiness for inferring the credits, where fairness is achieved by considering the incremental marginal contribution of one of the agents. These methods fail to interpret the impact of agent observation on decision-making or explicitly present how they cooperate with each other. Whereas glass-box models in MARL, e.g., mixture soft decision trees (Liu et al., 2022) and visual perception (Blumenkamp & Prorok, 2021), do not achieve exciting performance. To resolve these problems, we propose a novel interpretable value decomposition method in this paper.

Generalized Additive Models. GAMs are generally regarded as powerful inherently-interpretable models in the machine learning community (Hastie & Tibshirani, 1986). It independently learns a shape function for each feature and sums the outputs of these functions to obtain the final model prediction. Previous work (Lou et al., 2013) has found that standard forms of GAMs are limited in their representational power due to the absence of learning interactions between inherent features. As an improvement, Lou et al. (2013) proposed GA^{2}M that incorporates the complexity of pairwise interactions into GAMs. To improve stability and performance, different variants of shape functions in GAMs have been investigated, including deep neural networks (Agarwal et al., 2021), polynomial kernel models (Dubey et al., 2022), and oblivious decision trees (Chang et al., 2022). Further, NIT (Tsang et al., 2018) and pureGAM (Sun et al., 2022) reduce complexity by adding constraint terms, achieving increased interpretability. Our work falls under the umbrella of the GAM family. We are the first to develop GAMs in value-based MARL by utilizing them to disentangle the joint action-value function across different interactions, thereby obtaining intrinsic and interpretable higher-order shape functions of the agents.

F. Experimental Details

F.1. Benchmarks and Settings

In our paper, we introduce two types of testing benchmarks as shown in *Figure 9*, including Level Based Foraging (LBF) and StarCraft Multi-Agent Challenge (SMAC). In this section, we will describe the details and settings of these benchmarks.

(a) Level Based Foraging (b) StarCraft Multi-Agent Challenge

Figure 9. Two benchmarks used in our experiments.

Table 1. Experimental settings of Level Based Foraging.

HYPERPARAMETER	VALUE	DESCRIPTION
MAX PLAYER LEVEL	3	MAXIMUM AGENT LEVEL ATTRIBUTE
MAX EPISODE LENGTH	50	MAXIMUM TIMESTEPS PER EPISODE
BATCH SIZE	32	NUMBER OF EPISODES PER UPDATE
TEST INTERVAL	10,000	FREQUENCY OF EVALUATING PERFORMANCE
TEST EPISODES	32	NUMBER OF EPISODES TO TEST
REPLAY BATCH SIZE	5000	MAXIMUM NUMBER OF EPISODES STORED IN MEMORY
DISCOUNT FACTOR γ	0.99	DEGREE OF IMPACT OF FUTURE REWARDS
TOTAL TIMESTEPS	1,050,000	NUMBER OF TRAINING STEPS
START ε	1.0	THE START ε VALUE TO EXPLORE
FINISH ε	0.05	THE FINISH ε VALUE TO EXPLORE
ANNEAL STEPS FOR ε	50, 000	NUMBER OF STEPS OF LINEAR ANNEALING
TARGET UPDATE INTERVAL	200	THE TARGET NETWORK UPDATE CYCLE

Table 2. Introduction of scenarios in SMAC benchmark.

MAP NAME	ALLY UNITS	ENEMY UNITS	TOTAL TIMESTEPS	SCENARIO TYPE
$8m$	8 MARINES	8 MARINES	$2M$	EASY
$2s3z$	2 STALKERS, 3 ZEALOTS	2 STALKERS, 3 ZEALOTS	$2M$	EASY
$2s\_vs\_1sc$	2 STALKERS	1 SPINE CRAWLER	$2M$	EASY
$3s5z$	3 STALKERS, 5 ZEALOTS	3 STALKERS, 5 ZEALOTS	$2M$	HARD
$3s\_vs\_5z$	3 STALKERS	5 ZEALOTS	$2M$	HARD
$2c\_vs\_64zg$	2 COLOSSI	64 ZERGLINGS	$2M$	HARD
$5m\_vs\_6m$	5 MARINES	6 MARINES	$2M$	HARD
$8m\_vs\_9m$	8 MARINES	9 MARINES	$2M$	HARD
*MMM*2	1 MEDIVAC, 2 MARAUDERS, AND 7 MARINES	1 MEDIVAC, 3 MARAUDERS, AND 8 MARINES	$2M$	SUPER HARD
$3s5z\_vs\_3s6z$	3 STALKERS, 5 ZEALOTS	3 STALKERS, 6 ZEALOTS	$5M$	SUPER HARD
corridor	6 ZEALOTS	24 ZERGLINGS	$5M$	SUPER HARD
$6h\_vs\_8z$	6 HYDRALISKS	8 ZEALOTS	$5M$	SUPER HARD

Level Based Foraging. Christianos et al. (2020) first uses this implementation of LBF. This environment is a mixed game of cooperation and competition. Specifically, agents are placed in a 10×10 grid world and each one is assigned a level. The observation of an agent has a 5×5 field of view around it. Their goal is to eat food that is also randomly scattered. Only if the total level of the agents involved in eating is equal to or higher than the food level does the agents receive a positive reward, which is the normalized food level. Furthermore, we set the penalty reward for movement to -0.002. On this basis, we use two task instances with different configurations, of which one is 2 food with 4 agents, and 3 food with 3 agents. We give other experimental settings in *Table 1*.

StarCraft Multi-Agent Challenge. The SMAC (Samvelyan et al., 2019) is one of the most popular multi-agent environments to test the performance of MARL algorithms. All algorithm implementations are based on StarCraft II (SC2.4.10 version) unit micromanagement tasks, and note that results from different versions are not comparable. We set the built-in AI difficulty of all enemy units by configuring difficulty=7, and all allied units are controlled by the corresponding RL algorithm. The allies need to learn a series of strategies to defeat all the enemies and win within the specified exploration length. In this paper, we evaluate all algorithms on 12 challenging combat scenarios in SMAC, and *Table 2* presents a brief introduction of these scenarios and the maximum training step. Furthermore, the specific environmental settings adhere to the original setups, as described in *Table 3*.

F.2. Hyperparameters of Baselines

We compare our method against nine popular value-based baselines, including VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2018), QTRAN (Son et al., 2019), Qatten (Yang et al., 2020a), QPLEX (Wang et al., 2021), Weighted QMIX (mainly

Table 3. Experimental settings of StarCraft Multi-Agent Challenge.

HYPERPARAMETER	VALUE	DESCRIPTION
DIFFICULTY	7	ENEMY UNITS WITH BUILT-IN AI DIFFICULTY
BATCH SIZE	32	NUMBER OF EPISODES PER UPDATE
TEST INTERVAL	10,000	FREQUENCY OF EVALUATING PERFORMANCE
TEST EPISODES	32	NUMBER OF EPISODES TO TEST
REPLAY BATCH SIZE	5000	MAXIMUM NUMBER OF EPISODES STORED IN MEMORY
DISCOUNT FACTOR γ	0.99	DEGREE OF IMPACT OF FUTURE REWARDS
START ε	1.0	THE START ε VALUE TO EXPLORE
FINISH ε	0.05	THE FINISH ε VALUE TO EXPLORE
ANNEAL STEPS FOR EASY & HARD	50,000	NUMBER OF STEPS OF LINEAR ANNEALING ε
ANNEAL STEPS FOR SUPER HARD	100,000	NUMBER OF STEPS OF LINEAR ANNEALING ε
TARGET UPDATE INTERVAL	200	THE TARGET NETWORK UPDATE CYCLE

Table 4. The specific structure of the shape function.

NO.	STRUCTURE
1ST LAYER	[ABS(LINEAR.WEIGHT), LINEAR(ORDER NUMBER, 8), ELU]
2ND LAYER	[ABS(LINEAR.WEIGHT), LINEAR(8, 4), ELU]
3RD LAYER	[ABS(LINEAR.WEIGHT), LINEAR(4, 1)]

OW-QMIX, and we rename it WQMIX in our experiments) (Rashid et al., 2020), CDS[2] (Li et al., 2021a), DVD (Li et al., 2022a), and SHAQ[3] (Wang et al., 2022), whereas the implementation of baselines is based on PyMARL[4]. All hyperparameters follow the code provided by the authors, and are maintained at a learning rate of 0.0005 by the RMSprop optimizer. Note that the learning rate of SHAQ is fine-tuned to each different scenario, which is unfair to the other baselines, hence the hyperparameters are set identically to others.

F.3. Hyperparameters of NA^{2}Q

In this paper, we utilize a recurrent style local Q-network with its default hyperparameters, specifically, the individual Q-function $Q_i(\tau_i, u_i)$ contains a GRU layer with a 64-dimensional hidden state and a ReLU activation layer. The optimization for individual Q-functions is conducted using RMSprop with weight decay and a learning rate of 0.0005. Regarding the generative model G_ω, both encoder and decoder are comprised of two fully connected layers with a 32-dimensional hidden state, optimizing the learnable parameters by Adam with a learning rate of 0.0005. Additionally, we set the weight β of the loss to 0.1. In the mixing network, we employ a small dimensional MLP for each shape function f_k in order-1 and order-2, whose details are shown in *Table 4*. Finally, for the attention mechanism, we set the hidden layer size to 64 for $\boldsymbol{w}_s$ and $\boldsymbol{w}_z$.

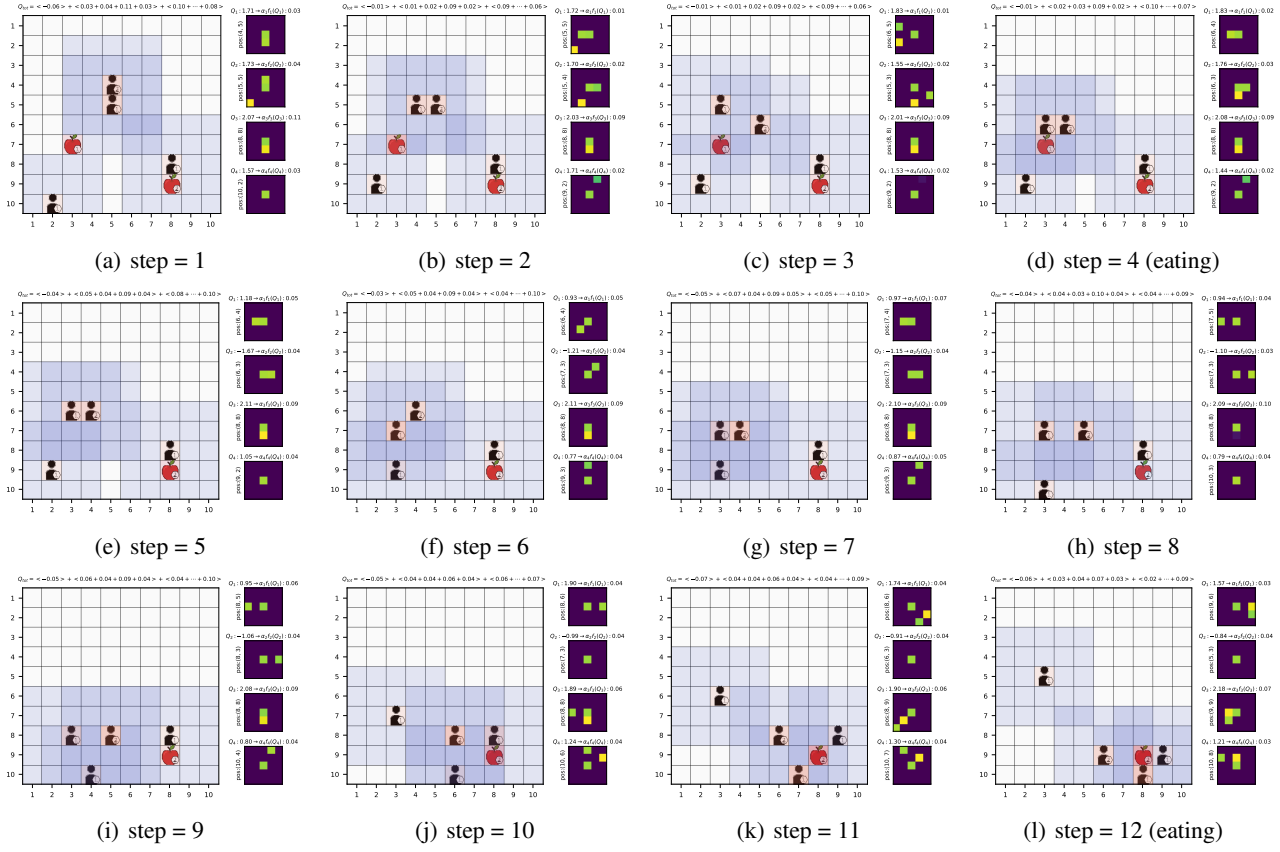
F.4. Infrastructure

Experiments are performed on an NVIDIA RTX 3080Ti GPU and an Intel I9-12900k CPU. We train our approach to run from 1 to 20 hours per scenario, depending on the complexity and length of the episode for each scenario.

G. Interpretability on LBF

Figure 10 demonstrates the contribution of agents and sub-teams on an episode in the LBF task, as well as showing the agent's corresponding mask. It is clear that NA^{2}Q accurately models the contribution of any agent or coalition of agents to the overall success. Furthermore, unsampled individual semantics can help us diagnose in a more interpretable way the relative importance of individual agent masks to relevant observations in the decision-making process.

[2]The code of CDS is from https://github.com/lich14/CDS.
[3]The code of SHAQ is from https://github.com/hsvgbkhgbv/shapley-q-learning.
[4]The source code of implementations is from https://github.com/oxwhirl/wqmix.

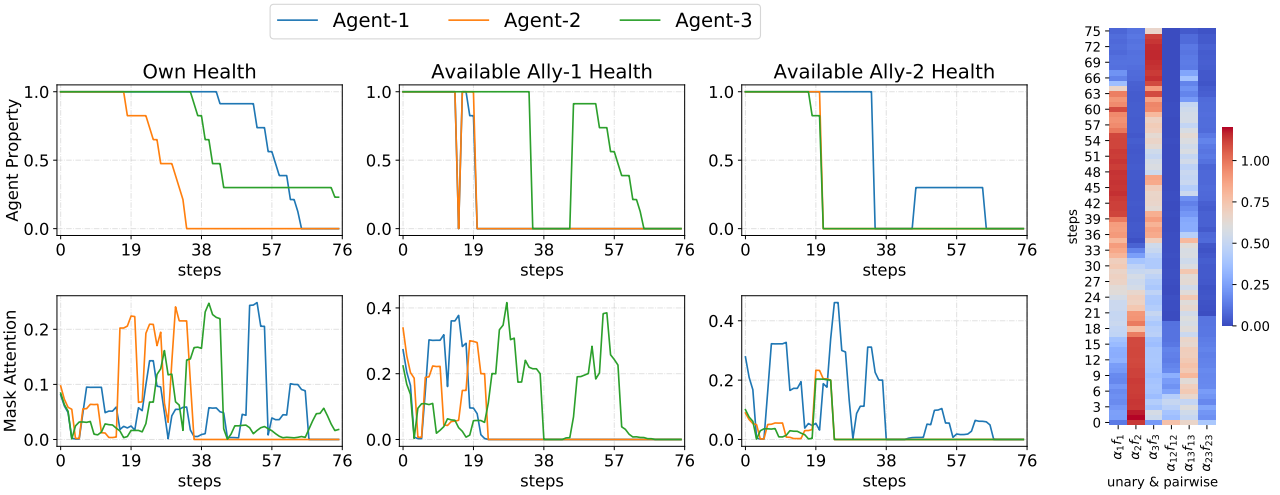


(a) step = 1 (b) step = 2 (c) step = 3 (d) step = 4 (eating)

(e) step = 5 (f) step = 6 (g) step = 7 (h) step = 8

(i) step = 9 (j) step = 10 (k) step = 11 (l) step = 12 (eating)

Figure 10. Visualization of the agent's mask on an episode, and the title indicates the contribution of each individual and agent alliance. The highlighted areas are the important regions for making decisions. As expected, when the environment changes, the attention and contribution of the agents also change accordingly.



(a) Agent properties and corresponding mask values (b) Agent contributions

Figure 11. Visualization of property semantics and agent contributions on the 3s_vs_5z scenario.

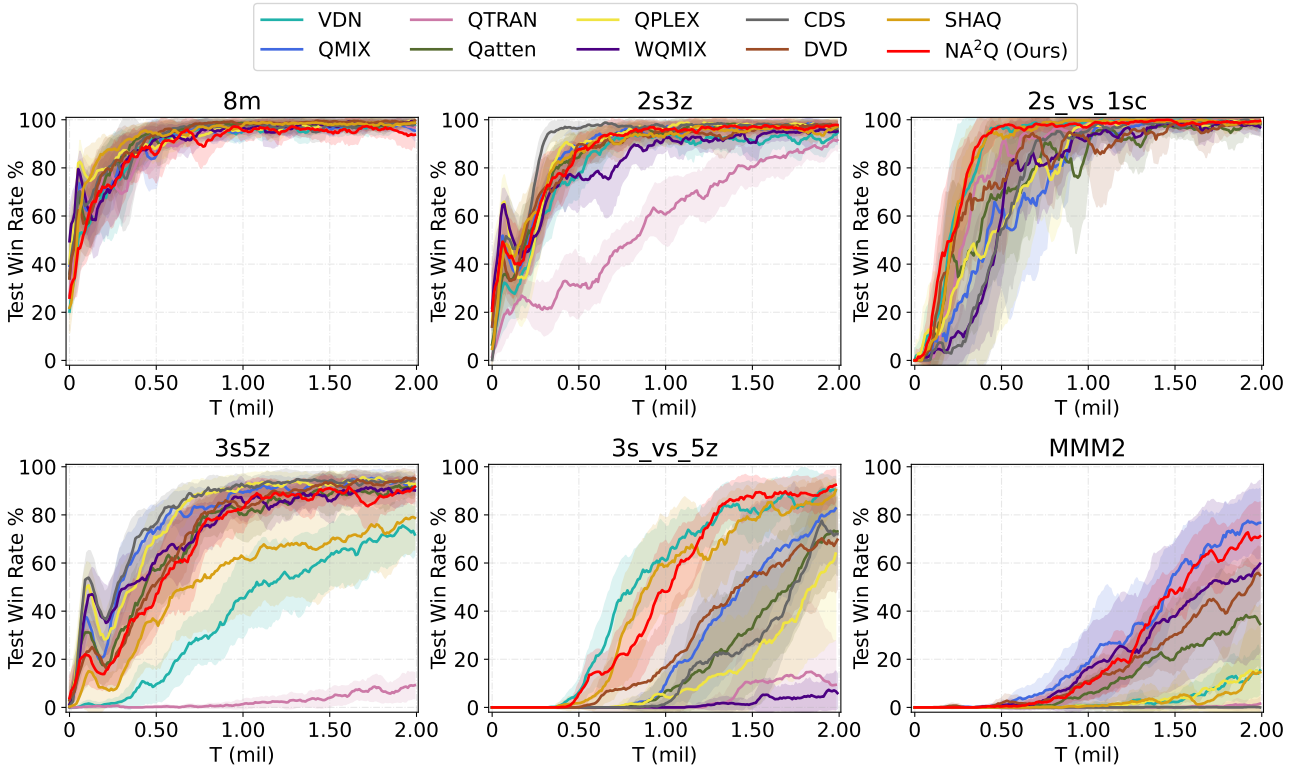


Figure 12. Test win rate % for six extra scenarios of SMAC benchmark.

H. Performance Results on Extra SMAC Maps

To thoroughly compare the performance of our method against the baselines, we experiment with six extra scenarios in *Figure 12* on SMAC, including 8m, 2s3z, 2s_vs_1sc, 3s5z, 3s_vs_5z, and MMM2. The parameter settings are in accordance with the previous experiments. It is obvious that NA^{2}Q still achieves impressive results on these six scenarios.

I. Additional Interpretability on SMAC

To further clarify the interpretability of NA^{2}Q, we select three properties related to the health of the agents to represent identity semantics, including own health, available Ally-1 health, and available Ally-2 health, and display the contribution of the corresponding agent on an episode. As shown in *Figure 11(a)*, the horizontal coordinate represents the number of steps on the episode, and the two vertical coordinates represent corresponding properties and semantic mask values, respectively. We find that the importance of the mask increases when the observed agent is harmed. Specifically, the teams are attacked with the sequence of Agent-2, Agent-3, and Agent-1, and the importance of their features peaked, respectively. Also, the corresponding mask is elevated when the visible ally receives damage. At the same time, we visualize the agent contributions to the unary and pairwise shape functions as shown in *Figure 11(b)*, where the steps increase from bottom to top and the horizontal ordination indicates the contribution id. The results show that the agents have different sensitivities at different stages of the battle. For example, Agent-2 performs a kiting operation, causing it to have a high contribution at the beginning stage. Meanwhile, Agent-1 and Agent-3 engage in cooperative attacks, resulting in higher contributions from sub-teams than from individual agents. In the later stages, agents are attacked separately, leading to higher contributions from individuals. Notably, the earlier death of Agent-2 leads to the pairwise shape functions associated with it remaining at depressed values. In summary, the NA^{2}Q can understand complex observations by diagnosing identity semantics and better explain the sub-spaces within order-2 interactions for the decomposition of the joint action-value function.