Weighted Mean-Field Multi-Agent Reinforcement Learning via Reward Attribution Decomposition

Tingyu Wu[1], Wenhao Li[1], Bo Jin[1], Wei Zhang[2], and Xiangfeng Wang[1]♠

[1] School of Computer Science and Engineering, East China Normal University, Shanghai, China
[2] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Sichuan, China
♠ Corresponding author: xfwang@cs.ecnu.edu.cn

Abstract. Existing MARL algorithms have low efficiency in many-agent scenarios due to the complex dynamic interaction when agents growing exponentially. Mean-field theory has been introduced to improve the scalability where complex interactions are approximated by those between a single agent and the mean effect from neighbors. However, only considering the averaged actions of neighborhood at last step and ignoring the dynamic influence of neighbors leads to unstable training procedures and sub-optimal solutions. In this paper, the Weighted **M**ean-**F**ield Multi-Agent Reinforcement Learning via **R**eward **A**ttribution **D**ecomposition (MFRAD) framework is proposed by differentiating heterogeneous and hysteresis neighbor effect with weighted mean-field approximation and reward attribution decomposition. The multi-head attention is employed to calculate the weights which formulate the weighted mean-field Q-function. To further eliminate the impact of hysteresis information, reward attribution decomposition is integrated to decompose weighted mean-field Q-value, improving the interpretability of MFRAD and achieving fully decentralized execution without information exchanging. Two novel regularization terms are also introduced to guarantee the consistency of temporal relationship among agents and unambiguity of local Q-value with no agents. Numerical experiments on many-agent scenarios demonstrate the superior performance against existing baselines.

Keywords: Multi-Agent Reinforcement Learning · Weighted Mean-Field Approximation · Reward Attribution Decomposition.

1 Introduction

In recent years, multi-agent reinforcement learning (MARL) has registered great potential in multi-agent systems, showing extraordinary performance in various scenarios, such as multi-player games [27], resource allocation [19,32], and network routing [13,14]. When the number of agents increases, classic MARL algorithms are far less effective than expectation due to the non-stationary issue [15] that agents not only interact with the environment but also with other

agents. Besides, the increasing action space resulting in the curse of dimensionality brings new challenges. Therefore, some efficient approaches [16,6] are proposed recently to solve above problems, especially the investigation of centralized training and decentralized execution (CTDE) framework [12,20,11,10].

However, many real-world scenarios [1,31] contain hundreds of agents cooperating and struggling with each other, bringing about the more difficult learning procedure due to the enormous action space and exponential dynamic interaction. The CTDE framework can not directly adapt to such many-agent scenarios due to the existence of centralized critic. [26] takes advantage of mean-field theory to realize scalability, transferring the many-agent interaction into the interactions between every ego agent and the approximated mean-field effect of the overall population. It should be noted that the mean-field approximation strongly relies on the introduction of the mean action of neighbors, which is directly averaged by all the neighbor actions, ignoring the different influence may caused by different locations, types or any attribute of neighbors. Unfortunately, most of derivative works [5,9,3] based on mean-field theory either focus on determining the accurate accessible neighbors or simply extend the same type agents to multi-types. Though weighted information [2,18,25] has also been considered, they still formulate the mean-field function with hysteresis information that same as original mean-field MARL, i.e., generating the mean action from the last step. The theoretical idea or the empirical results [4,18] indicate that, the miscalculation of the mean-field effect caused by equal treatment of neighbors and use of hysteresis information may leads to the wrong direction of optimization, which matches with the massive oscillations during training.

In order to eliminate the unexpected effect of mean action in mean-field approximation, we first introduce attention mechanism, similar with existing methods [2,25], to differentially process neighbors' information. Then we decompose the weighted local Q-value via reward attribution decomposition which is inspired by [29], formulating the weighted mean-field approximation as a joint optimization over an implicit reward assignment among the ego agent and its neighbors. After decomposition, not only in the training phase, the ego agent can better distinguish the impact of the neighbors' hysteresis information, but also the execution phase is now fully decentralized without any information exchanging, especially dropping the effect from hysteresis information utilization. Moreover, it distinguishes from the value decomposition methods [22,17] owing to the latter is decomposed from team perspective which only adapted in cooperative settings, while the former from individual point of view under the guidance of different reward assignment, improving the interpretability to some extent.

To this end, we proposed the Weighted **M**ean-**F**ield MARL via **R**eward **A**ttribution **D**ecomposition (MFRAD) framework by differentiating heterogeneous and hysteresis neighbor effect with weighted mean-field approximation and reward attribution decomposition. Specifically, we first achieve weighted mean-field approximation by calculating the weighted state-action embeddings of neighbors nearby the ego agent. Then, in reward attribution decomposition, considering the effect of the ego agent's interaction with its neighbors caused by

its own actions and neighbors' actions, the pairwise local Q-function is decomposed as two terms: the SELF-term that only relies on the agent's own state, and the neighbor term that is related to the weighted mean-field effect of other agents which combined with multi-head attention mechanism. Intuitively, the relationship between agents maintains temporarily stable even if the characteristics of the neighbor agents change at a certain moment. Thus we propose a novel regularization term named temporal relationship regularization to maintain the temporary difference of attention weights between timesteps. Moreover, decomposing pairwise local Q-function with a simple addition, the solution of two terms might not be unique. Drawing the inspiration from previous work [29], we introduce an extra regularization term to guarantee the unambiguity of ego agent's local Q-value with no neighbors. Main contributions are listed as follows: 1) We propose the weighted mean-field approximation that captures the fine-grained neighbor information and employ multi-head mechanism to calculate the dynamic mean-field effect; 2) The idea of reward attribution decomposition is introduced to reduce the negative effect of antique signal from calculating the delayed mean action of neighbors, transforming the Q-function of each agent into summation of local Q-function and weighted mean-field Q-function that related to neighbors from individual perspective; 3) Multiple many-agent experiments on MAgent and CityFlow are conducted to verify the proposed MFRAD algorithm can achieve higher return and stable performance in both cooperative and competitive tasks, and has certain scalability in real-world scenarios where cooperation and struggle coexist.

2 Related Work

Mean-field Games. Introducing mean-field theory into MARL has gained wide attention recently, which approximates the complex interactions between agents into the interaction between ego agent and the neighboring agent distribution [7], eliminating the dimensional disaster. It also effectively alleviates exploration noise caused by multiple agents so that each agent can efficiently make beneficial local decisions. [26] firstly proposes a model-free scheme for learning the optimal action based on mean-field theory. [4] relaxes the assumption on the neighbor range in [26], establishing the mean-field effect of accessible agents which in a predefined observation range or visible distribution. When it comes with more complex game settings, [3] approximates the joint action of N agents to N mean actions while [25] approximately estimates the inter-type and intra-type interactions between agents without exact number. Also, weighted information [25] and graph neural network with attention mechanism [8] has been introduced to model the neighbor relationship, while existing works mainly focus on the weighted action distribution to formulate the pairwise mean-field Q-function directly.

Value Function Decomposition. The most straightforward way to train a MARL task is to learn each agent's Q-function independently[23], while it ignores

the dynamic influence of other agents that leads to the non-stationary environment especially in many-agent scenarios. Value function decomposition(VFD) methods, e.g., VDN[22], QMIX[17], QTRAN[21], adopt CTDE paradigm to rewrite the joint Q-function as $Q^{\pi}(s, \mathbf{a}) = \phi\left(s, Q^1\left(o^1, a^1\right), \ldots, Q^N\left(o^N, a^N\right)\right)$ where the formulation of ϕ differs in each method. Although these VFD methods successfully solve the non-stationary issue, none of them is well adapted to many-agent scenarios where large-scale numbers of agent exist.

3 Preliminaries

3.1 Markov Decision Process and Markov Game

The Markov Game with N agents which generalizes from Markov Decision Process is formalized by the tuple $\Gamma \triangleq \left(\mathscr{S}, \mathscr{A}^1, \ldots, \mathscr{A}^N, r^1, \ldots, r^N, p, \gamma\right)$, where $\mathscr{S}$ represents the state space and $\mathscr{A}^j$ denotes the actions of the agent $j \in \{1, \ldots, N\}$. The reward function is $r^j : \mathscr{S} \times \mathscr{A}^1 \times \cdots \times \mathscr{A}^N \to \mathbb{R}$. All agents maximize their discounted sum of rewards with the discount factor $\gamma \in [0, 1)$. p is the transition probability $\mathscr{S} \times \mathscr{A}^1 \times \cdots \times \mathscr{A}^N \to \Omega(\mathscr{S})$ where $\Omega(\mathscr{S})$ is the collection of state space probability distributions. For agent j, the corresponding policy is defined as $\pi^j : \mathscr{S} \to \Omega\left(\mathscr{A}^j\right)$ and each agent is trying to maximize its return over the consideration of others' behaviors, where $\Omega\left(\mathscr{A}^j\right)$ represents the set of probability distributions on the agent's j action space $\mathscr{A}^j$. The joint policy of all agents can be denoted as $\boldsymbol{\pi} \triangleq \left[\pi^1, \ldots, \pi^N\right]$. Considering the initial state s, the value function of agent j under the joint policy $\boldsymbol{\pi}$ is formulated as the expected future cumulative discount reward: $v_{\pi}^j(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\boldsymbol{\pi}, p}\left[r_t^j \mid s_0 = s, \boldsymbol{\pi}\right]$. The Q-function of agent j under the joint policy $\boldsymbol{\pi}$ can be formalized as: $Q_{\pi}^j(s, \boldsymbol{a}) = r^j(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim p}\left[v_{\pi}^j\left(s'\right)\right]$, where s' represents the next state.

3.2 Mean-field Reinforcement Learning

Mean-field MARL approximates the complicated interactions in many-agent scenarios into the bilateral estimation of two agents where the second agent corresponds to the mean effect of the overall population. The Q-function $Q^j(\boldsymbol{s}, \boldsymbol{a})$ in mean-field MARL will be decomposed by using only local bilateral interactions:

$$Q^j(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{N^j} \sum_{k \in \mathcal{N}(j)} Q^j\left(\boldsymbol{s}, a^j, a^k\right), \tag{1}$$

where $\mathcal{N}(j)$ represents the sequence number set of agent j's neighbors with size $N^j = |\mathcal{N}(j)|$. After decomposing the Q-function through the bilateral estimation of the agent and its neighbors, it dramatically reduces the interaction complexity in the large-scale scenarios. So this decomposition converts the joint Q-function into the mean field formulation $Q_{\mathrm{MF}}^j\left(\boldsymbol{s}, a^j, \bar{a}^j\right)$ where the mean action $\bar{a}^j$ is calculated according to the neighboring agent set $\mathcal{N}(j)$. Considering the small disturbance, a^k is denoted as : $a^k = \bar{a}^j + \delta a^{j,k}$, where $\bar{a}^j = \frac{1}{N^j} \sum_{k \neq j} a^k$. The Q-function is updated in a recurrent manner:

$$Q_{t+1}^j\left(\boldsymbol{s}, a^j, \bar{a}^j\right) = (1 - \alpha) Q_t^j\left(\boldsymbol{s}, a^j, \bar{a}^j\right) + \alpha\left[r^j + \gamma v_t^j\left(\boldsymbol{s}'\right)\right], \tag{2}$$

where α is the learning rate and r^j is the obtained reward. $\boldsymbol{s}$ and $\boldsymbol{s}'$ represents the old state and resulting state respectively. The value function $v_t^j\left(s'\right)$ for agent j at time t is formulated as:

$$v_t^j\left(\boldsymbol{s}'\right)=\sum_{a^j}\pi_t^j\left(a^j \mid \boldsymbol{s}', \bar{a}^j\right)\mathbb{E}_{\bar{a}^j\left(\boldsymbol{a}^{-j}\right)\sim\boldsymbol{\pi}_t^{-j}}\left[Q_t^j\left(\boldsymbol{s}', a^j, \bar{a}^j\right)\right], \tag{3}$$

with $\bar{a}_t^j=\frac{1}{N^j}\left(\sum_{k\neq j}a_t^k\right)$, $a_t^k\sim\pi^k\left(\cdot \mid \boldsymbol{s}_t, \bar{a}_{t-1}^k\right)$, and

$$\pi_t^j\left(a_t^j \mid \boldsymbol{s}_t, \bar{a}_{t-1}^j\right)=\frac{\exp\left(-\beta Q^j\left(\boldsymbol{s}_t, a_t^j, \bar{a}_{t-1}^j\right)\right)}{\sum_{a_t^{j'}\in A^j}\exp\left(-\beta Q^j\left(\boldsymbol{s}_t, a_t^{j'}, \bar{a}_{t-1}^j\right)\right)}, \tag{4}$$

where β is the Boltzmann parameter and π denotes the Boltzmann policy .

4 Algorithm

In this section, we introduce the proposed MFRAD algorithm illustrated in Fig.1. Considering the limitations of mean-field MARL, we firstly extend the existing mean-field approximation to the form with weight information and give the detailed mathematical derivation. Secondly, inspired by [29], we transform the joint Q-function into the integration of ego agent's individual Q-function and weighted mean-field Q-function of its neighbors, which called *reward attribution decomposition*, utilizing the multi-head attention to calculate the weights.

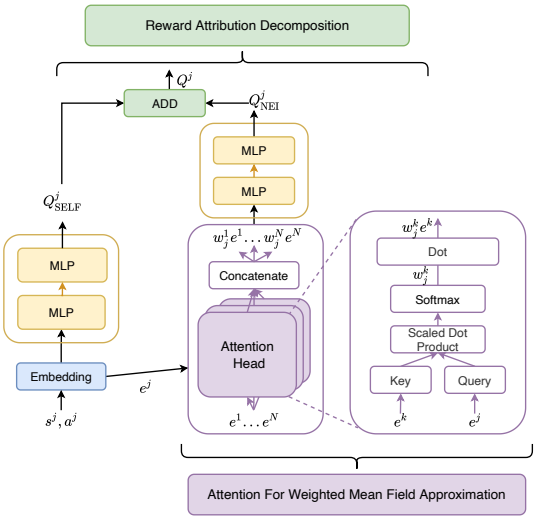


Fig. 1. Architecture of MFRAD. Each agent calculate its Q^j_{SELF} based on state-action embeddings which consists of local observation and action. Meanwhile, multi-head attention module receives the state-action embeddings of ego agent's neighbors as input, calculating attention weights as mean-field weights to construct the weighted mean-field effect Q^j_{NEI}. Finally, these two items constitute the decentralized Q^j.

4.1 Weighted Mean-Field Approximation

Drawing inspiration from existing works [2,18], we rewrite the original mean-field approximation formula (1) into a form with weight information

$$Q^j(\boldsymbol{s},\boldsymbol{a}) = \sum_{k\in\mathcal{N}(j)} w_j^k Q^j\left(s^j,a^j,s^k,a^k\right), \tag{5}$$

where w_j^k represents the weight of each neighbor's effect on ego agent j, and $0 \le w_j^k \le 1$, $\sum_{k\in\mathcal{N}(j)} w_j^k = 1$. $\mathcal{N}(j)$ represents the sequence number set of agent j's neighbors with size $N^j = |\mathcal{N}(j)|$. As for clarity, we denote $e^j \triangleq e^j\left(s^j,a^j\right)$ which performed with embedding operation, thus (5) can be reformulate as

$$Q^j(\boldsymbol{s},\boldsymbol{a}) = \sum_{k\in\mathcal{N}(j)} w_j^k Q^j\left(e^j,e^k\right). \tag{6}$$

Similar as the deviation in mean-field approximation, the weighted mean-field approximation is still based on the weighted average effect of the state-action pair $\bar{e}^j$ from the adjacent agent set $\mathcal{N}(j)$. We represent the local state-action information of each neighbor as the sum of weighted average effect $\bar{e}^j$ and a small disturbance $\delta e^{j,k}$, that is, $e^k = \bar{e}^j + \delta e^{j,k}$, where $\bar{e}^j = \sum_{k\in\mathcal{N}(j)} w_j^k e^k$ can be interpreted as neighborhood state-action distribution. Then according to Taylor's theorem, if the bilateral weighted Q-function of agent k is twice-differentiable, then (6) can be expanded as

$$\begin{aligned} Q^j(\boldsymbol{s},\boldsymbol{a}) = \sum_{k\in\mathcal{N}(j)} w_j^k Q^j\left(e^j,e^k\right) = \sum_{k\in\mathcal{N}(j)} w_j^k \Big[& Q^j\left(e^j,\bar{e}^j\right) + \\ & \nabla_{\bar{e}^j} Q^j\left(e^j,\bar{e}^j\right)\delta e^{j,k} + \underbrace{\frac{1}{2}\delta e^{j,k}\cdot\nabla^2_{\bar{e}^{j,k}} Q^j\left(e^j,\bar{e}^{j,k}\right)\delta e^{j,k}}_{R^j_{e^j}\left(e^k\right)}\Big], \end{aligned} \tag{7}$$

where the first term is merged as $Q^j\left(e^j,\bar{e}^j\right) = \sum_{k\in\mathcal{N}(j)} w_j^k Q^j\left(e^j,\bar{e}^j\right)$, and the second term equals to zero since $\bar{e}^j = \sum_{k\in\mathcal{N}(j)} w_j^k e^k$. In addition, $R^j_{e^j}\left(e^k\right)$ is the Taylor polynomial's remainder where $\bar{e}^{j,k} = \bar{e}^j + \epsilon^{j,k}\delta e^{j,k}, \epsilon^{j,k} \in [0,1]$, so (7) is finally reduced to

$$Q^j(\boldsymbol{s},\boldsymbol{a}) \approx Q^j\left(e^j,\bar{e}^j\right) = Q^j\left(s^j,a^j,\sum_{k\in\mathcal{N}(j)} w_j^k s^k,\sum_{k\in\mathcal{N}(j)} w_j^k a^k\right). \tag{8}$$

Therefore, based on the weighted mean effect, the bilateral interaction between agent j and its neighbor agent k is simplified as the local pairwise interaction between the ego agent and the mean-field agent, and the latter is abstracted from the weighted mean effect of neighborhood state-action information.

4.2 Reward Attribution Decomposition

Though weighted information is introduced, the another drawback of mean-field MARL that the mean-field effect of neighbor is generated from obsolete information, which is unreasonable to choose actions according to the generated policy. Drawing inspiration from [29], we decompose the Q-value of ego agent

into its own part and that of neighbor agent via reward assignment mechanism. Therefore, we realize the decentralized execution without historical information sharing among agents, alternatively, the weighted mean-field effect of neighbors is dexterously converted into the centralized training process. The intuition of this decomposition is that the effect of the ego agent's interaction with its neighbors is caused by two factors, that is, the action taken by the ego agent based on its local observation and the actions taken by neighbors based on their local observations, and all these actions are chosen under the guidance of reward assigning. We will explain in more detail later why the proposed MFRAD is able to achieve fully decentralization in execution. As a result, the weighted mean-field Q-value for each agent can be effectively decomposed, which decoupled the description of global information under the partially observed assumption. It also realizes high scalability in many-agent scenarios from individual perspective, making up for the limitation of mean-field MARL in calculating mean action of neighborhood during the decentralized execution phase. Specifically, according to the weighted mean-field approximation (8), we have

$$\begin{aligned} Q^j(\boldsymbol{s}, \boldsymbol{a}) &= Q^j\left(s^j, a^j, \sum_{k \in \mathcal{N}(j)} w_j^k s^k, \sum_{k \in \mathcal{N}(j)} w_j^k a^k\right) \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^j \left(s_t^j, a_t^j, \sum_{k \in \mathcal{N}(j)} w_j^k s_t^k, \sum_{k \in \mathcal{N}(j)} w_j^k a_t^k\right) \middle| s_0 = \mathbf{s}, a_0 = \mathbf{a}\right], \end{aligned} \tag{9}$$

followed with the principle of reward attribution decomposition that explained in [29], agent acts according to a state not only because the reward to itself, but also because it is more rewarding than other agents. Therefore, we split the reward of an agent from the ego agent's point of view, that is, the reward is not only derived from itself, but also influenced by that of neighbors.

$$r^j\left(s_t^j, a_t^j, \sum_{k \in \mathcal{N}(j)} w_j^k s_t^k, \sum_{k \in \mathcal{N}(j)} w_j^k a_t^k\right) = r^j\left(s_t^j, a_t^j\right) + r^j\left(\sum_{k \in \mathcal{N}(j)} w_j^k s_t^k, \sum_{k \in \mathcal{N}(j)} w_j^k a_t^k\right), \tag{10}$$

then $Q^j(\boldsymbol{s}, \boldsymbol{a})$ can be further decomposed

$$\begin{aligned} Q^j(\boldsymbol{s}, \boldsymbol{a}) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^j\left(s_t^j, a_t^j\right) \middle| s_0 = \mathbf{s}, a_0 = \mathbf{a}\right] \\ &\quad + \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^j \left(\sum_{k \in \mathcal{N}(j)} w_j^k s_t^k, \sum_{k \in \mathcal{N}(j)} w_j^k a_t^k\right) \middle| s_0 = \mathbf{s}, a_0 = \mathbf{a}\right] \\ &\approx Q_{\mathrm{SELF}}^j(s^j, a^j) + Q_{\mathrm{NEI}}^j\left(\{w_j^k, s^k, a^k\}_{k \in \mathcal{N}(j)}\right). \end{aligned} \tag{11}$$

Finally, the weighted average Q-function is transformed into the summation of the local Q-function Q_{SELF} of the ego agent and the neighbor agent Q-function Q_{NEI} with weighted information.

4.3 Network Architecture

The overall architecture of the proposed MFRAD is illustrated in Figure 1. As discussed above, the local Q-function of each agent j consists of two parts: SELF-Q network and NEI-Q network. For detail, on the one hand, SELF-Q network parameterized by θ^j_{self} is separated for calculating the Q^j_{SELF} of each agent j based on its local observation s^j and action a^j. Noted that for each agent j, its local observation and action pair (s^j, a^j) is encoded as e^j via a state-action encoder[3] before fed into SELF-Q network to reduce the noise redundancy of original coarse information. On the other hand, the NEI-Q network parameterized by θ^j_{nei} employs the multi-head attention module (will be explained soon) to model the weighted neighbour state-action distribution, which is further processed for calculating the weighted effect Q^j_{NEI} of neighbors.

Concretely, the multi-head attention module introduces the attention mechanism to calculate the weights in the weighted mean-field approximation. The embedding vector e^j is regarded as the *query* vector, while e^k which contains neighbour state-action information is regarded as the *key* vector. The mean-field weights are then calculated by comparing the key vectors and query vector in terms of their dot similarity, which is evaluated through a softmax function

$$w^k_j \propto \exp\left(e^j \left(s^j, a^j\right)^T W^T_{\text{key}} W_{\text{query}} e^k \left(s^k, a^k\right)\right), \tag{12}$$

and we denote the parameters of attention module $(W_{\text{key}}, W_{\text{query}})$ of each agent j as θ^j_{att}.

Rather than the single-head attention, here we use multi-head attention mechanism to comprehensively utilize all aspects of the information of the agent from multiple angles and extract more abundant feature representation. Each attention head corresponds to a separate set of weight parameters $(W^m_{\text{key}}, W^m_{\text{query}})$ where $m \in [M]$ and M is the number of heads. For clarity, $(W^m_{\text{key}}, W^m_{\text{query}})$ is denoted as $\theta^{j,m}_{\text{att}}$ and $\Theta^j_{\text{att}} := \{\theta^{j,m}_{\text{att}}\}^M_{m=1}$ refers to parameters of all attention heads. The final weights are then obtained by averaging the attentions of multiple attention heads. That is, the observation-action distribution of neighbors e^{-j} is calculated as follows

$$e^{-j} = \frac{1}{M} \sum_m^M \sum_k w^{k,m}_j e^k. \tag{13}$$

The obtained e^{-j} is then fed into the NEI-Q network, and the Q-value of neighbors after the fusion of attention mechanism is calculated, which describes the comprehensive effect of neighbors on the ego agent with better explanation and representation. Finally, we perform the summation operation on Q^j_{SELF} and Q^j_{NEI} to formulate the $Q^j(\boldsymbol{s}, \boldsymbol{a})$.

4.4 Overall Optimization Objective

Intuitively, the interaction between agents is not a transient process, the relationship between agents maintains temporarily stable even if the characteristics of

[3] Without causing confusion, we incorporate the parameters of this encoder into θ^j_{self}.

the neighbor agent change at a certain moment. Therefore, the attention weight distribution should also remain stable in a short period of time. We use KL divergence to measure the difference of attention weights between timesteps as in [8], in order to keep the consistency of the temporal relationship. Therefore, the regularization term about temporal relationship, called temporal relationship regularization (TRR), is added to the loss function

$$\Omega(\theta^j_{\text{self}}, \Theta^j_{\text{att}}; s^l, a^l, s^{l,'}, a^{l,'}) = \frac{1}{\text{M}} \sum_{m=1}^{\text{M}} D_{\text{KL}} \left[w_j^{k,m}(s^l, a^l; \theta^j_{\text{self}}, \theta^{j,m}_{\text{att}}) \| w_j^{k,m}(s^{l,'}, a^{l,'}; \theta^j_{\text{self}}, \theta^{j,m}_{\text{att}}) \right], \tag{14}$$

where $l := \mathcal{N}(j) \cup j$, (s^l, a^l) and $(s^{l,'}, a^{l,'})$ are state-action pairs at two consecutive timesteps, $D_{\text{KL}}[\cdot\|\cdot]$ denotes the KL-divergence operator.

Moreover, revisiting (11), with a simple addition, the solution of Q^j_{SELF} and Q^j_{NEI} might not be unique. Indeed, we might add any constant to Q^j_{SELF} and subtract that constant from Q^j_{NEI} to yield the same local Q-value Q^j. Drawing the inspiration from previous work [29], we introduce an extra regularization term, Q^j_{NEI}. Intuitively, we hope that during the training process, Q^j_{NEI} can gradually converge to 0, so that there is a unique optimal solution for two terms in (11). From another perspective, the introduction of this regularization term is also similar to the teacher-student framework in transfer learning [28]. As learning progresses, Q^j_{NEI} gradually distills knowledge into Q^j_{SELF}. This enables the ego agent to adaptively process hysteresis information from neighbors. Further, this also enables MFRAD to only make decisions based on Q^j_{SELF} during the execution phase, without the need to communicate with the neighbors to calculate the average actions of them, enabling fully decentralized execution. Compared with the existing work based on mean-field theory, MFRAD has better scalability. We also observe that with NEI objective, training is much stabilized in following numerical experiments. By the way, since the occurance that no agents exists in the neighborhood may create ambiguity, this regularization term also making the guarantee that $\arg\max_{a^j} Q^j = \arg\max_{a^j} Q^i_{\text{SELF}}$.

In order to make MFRAD have faster convergence speed and better scalability, the parameters of all agents are shared. Therefore we denote the parameters of SELF-Q network, NEI-Q network, and multi-head attention module of any agent as θ_{self}, θ_{nei} and Θ_{att} respectively. Finally, the overall optimization objective for each agent j that integrates the regularization terms is shown as follows:

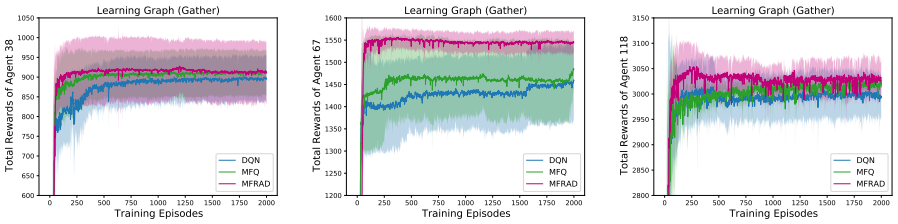
$$\begin{aligned} \text{L}(\theta_{\text{self}}, \theta_{\text{nei}}, \Theta_{\text{att}}) = \frac{1}{N} \sum_{j=1}^{N} \mathbb{E}_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{s'},\boldsymbol{a'}} \Big[\underbrace{\left(y^j - \left(Q^{\text{SELF}}_{\theta_{\text{self}}}\left(s^j, a^j\right) + Q^{\text{NEI}}_{\theta_{\text{nei}}}\left(s^k, a^k\right) \right) \right)^2}_{\text{DQN Objective}} \\ + \underbrace{\lambda_1 \left(Q^{\text{NEI}}_{\theta_{\text{nei}}}\left(s^j, a^j\right) \right)^2}_{\text{NEI Objective}} + \underbrace{\lambda_2 \Omega(\theta_{\text{self}}, \Theta_{\text{att}}; s^l, a^l, s^{l,'}, a^{l,'})}_{\text{TRR Objective}} \Big], \end{aligned} \tag{15}$$

where $k \in \mathcal{N}(j)$, $l := \mathcal{N}(j) \cup j$ and λ_1 and λ_2 represent the relative importance of two regularization terms against the optimization direction of original Q-function respectively.

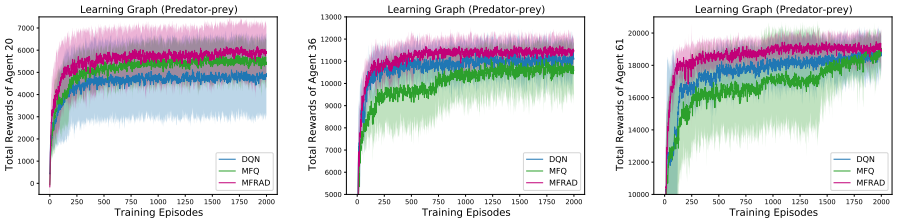
5 Experiments

We consider many-agent scenarios in this section and evaluate the performance of the proposed MFRAD framework. Firstly, three different tasks will be discussed based on MAgent [30] platform, including a competitive task (i.e., *Gather Game*), a cooperative task (i.e., *Predator-prey Game*) and a mixed cooperative-competitive task(i.e., *Battle Game*). Additionally, more detailed analysis on the *Battle Game* will be conducted to verify the effects of attention mechanism and two regularization terms. Moreover, we choose a real-world task on traffic flow to demonstrate the scalability of MFRAD which outperforms both existing rule-based and value function decomposition methods.

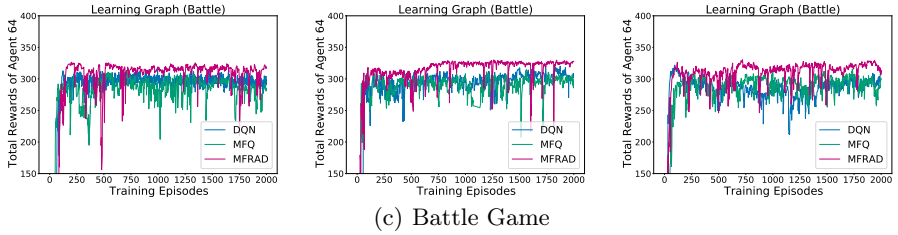
5.1 Results and Analysis



(a) Gather Game



(b) Predator-prey Game

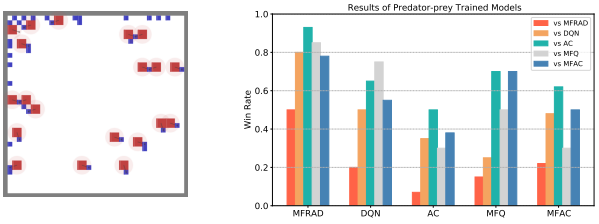


(c) Battle Game

Fig. 2. In all the learning graphs of three MAgent games with different agent numbers, MFRAD shows the best performance.

Experiments on Gather Game. In this scenario, agents compete for limited food resources as much as possible and can kill other agents to maximize their own survival time. The average total rewards with a standard deviation of 20 experiments for each algorithm is proposed in Fig. 2(a) where the agent number scales. MFRAD consistently outperforms with higher total reward. It is similar to other algorithms when in small scale, while gradually has faster convergence speed than MFQ and DQN nearly 100 episodes though starts with relatively slow growth rate.

Experiments on Predator-prey Game. Predator-prey Game is a fully cooperative task where the predators cooperate to capture as many preys as possible. As shown in Fig. 2(b), MFRAD estimates the influence of other agents accurately through the reward attribution mechanism when number of agent increasing. However, MFQ is inferior to MFRAD and DQN in terms of convergence speed and total reward, related to approximating the mean-field effect among indiscriminate neighbors. In addition, we visualize the pursuit process in Fig. 3(a) and find that MFRAD predator cooperate to capture alone rather than continue to chase preys that have been observed and chased by others. Then, we record the pursuing results of different methods which fight with each other for 200 episodes in Fig. 3(b). Obviously, MFRAD with weighted neighbor information always defeats other methods.



(a) (b)

Fig. 3. Results in testing phases. (a)In the Predator-prey Game, each agent cooperate to catch preys as much as possible. (b)Win rate among MFRAD and other baselines.

Experiments on Battle Game. The learning graph of Battle Game in which red army of 64 agents fight with the similar blue army is shown in Fig. 2(c). Since only the total reward corresponding to the red army is recorded when it is higher than that of the blue army, the timestep of each record in each round cannot be aligned synchronously. Therefore, calculating the mean and standard deviation of the total reward lacks physical meaning and cannot reflect the actual performance. We randomly select 3 results from 50 experiments to show the robustness of each algorithm to different random seeds. Although the convergence speed of MFRAD in the initial stage is slightly slower, it outperforms MFQ and DQN with higher cumulative reward and minor variance after convergence, showing the stable performance in multiple trials. This phenomenon

proves that the introduction of weighted mean-field effect after decomposition by reward has a significant impact on the stability of the training process.

In addition, the average individual reward in 2000 episodes and number of opponents killed by each algorithm are shown in Table. 1. Noted that we add MFAC and AC algorithm based on actor-critic architecture to enrich comparison information, however, MFRAD performs significantly better than AC-family methods. Besides, Multi-head attention mechanism plays an significant role in

Method	Agent36 vs 36		Agent 64 vs 64		Agent 196 vs 196		Agent 324 vs 324	
	Reward	Kill	Reward	Kill	Reward	Kill	Reward	Kill
DQN	0.13	35.31	0.18	62.9	0.09	193.43	0.08	**316.99**
AC	0.14	24.19	0.12	43.94	0.11	149.1	0.07	247.27
MFQ	0.14	35.21	0.12	62.92	0.09	193.3	0.08	315.32
MFAC	**0.15**	14.31	0.06	44.21	0.11	153.59	0.08	232.13
MFRAD	0.14	**36.47**	**0.19**	**63.3**	**0.12**	**194.97**	**0.09**	316.91

Table 1. Mean agent reward and number of opponents killed in different scales.

describing how important effect the neighbors make on the ego agent. From the ablation results in Table. 2, the proposed MFRAD benefits from multi-head attention to enrich the neighborhood information, proving the benefit of approximating weighted mean-field effect of neighbors. Thus, MFRAD converges to a higher total reward and remains the stable value than other algorithms.

Metric		Method		
		w/o attention	single-head attention	multi-head attention
Training	total reward	290.47	298.13	**302.62**
	individual reward	0.12	0.18	**0.19**
	kill number	58.4	62.9	**63.3**
Testing	total reward	216.1	239.74	**243.13**
	kill number	39.0	45.2	**48.9**

Table 2. Impact of Attention Mechanism.Mean reward and number of opponents killed is evaluated on both training and testing phases to show the influence of attention.

We also study the importance of NEI Loss and TRR loss by removing it from MFRAD in scenarios where both army contains 196 agents, as shown in Table. 3. Using these two regularization terms boosts the performance and stabilizes the training process, consistent with the proposed reward attribution decomposition.

Experiments on Traffic Flow Control. In order to investigate the scalability of MFRAD to more complex real-world task, we choose the traffic flow control task and experiment on the large-scale traffic flow platform named CityFlow. Following existing studies, we model each intersection as an RL agent and realize the communication by sharing information among agents. It is a typical mixed cooperative-competitive scenario.

We compare our model with the following two categories of methods: rule-based method(i.e., *Max-Pressure*[24]) and RL methods (i.e., *IQL,* QMIX, *VDN*)

Method	NEI objective	TRR objective	Battle Scenario Metrics	
			Mean Reward	Kill-Death Ratio
OL			887.92	1.88
OL-NEI	√		925.81	2.13
OL-NEI-TRR	√	√	**928.49**	**2.26**

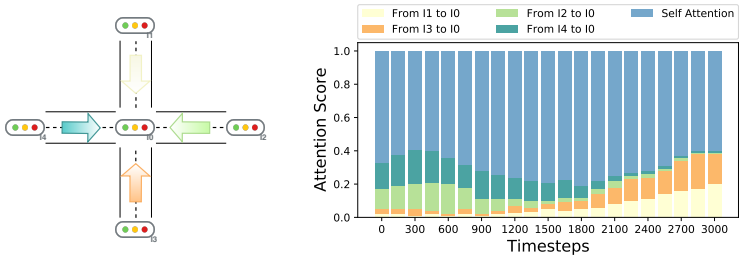
Table 3. Impact of regularization term of loss function. Mean total reward and kill-death ratio is evaluated to show that MFRAD with refactoring loss is better at killing opponents and protecting themselves.

based on value function decomposition. The average travel time, which calculates the average travel time of all the vehicles spent from waiting in the queue and leaving the intersection, is chosen to evaluate the performance of different methods. MFRAD achieves consistent performance improvements on both synthetic data and real-world data, it costs less travel time not only compared with rule-based method but also RL(VFD) methods.

Model	Arterial $_{1\times6}$	Grid $_{6\times6}$*bi*	Grid $_{6\times6}$*uni*	NewYork $_{16\times3}$	Hangzhou $_{4\times4}$
Max-Pressure	122.98	204.72	186.06	405.69	431.53
IQL	143.95	269.18	244.39	254.93	472.38
QMIX	110.27	542.63	678.27	226.15	562.39
VDN	131.53	468.94	630.82	198.24	358.73
MFRAD	**72.98**	**171.51**	**168.25**	**183.82**	**310.07**

Table 4. Performance on synthetic data and real-world data w.r.t average travel time

Ablation study is conducted to further analyze the effect of attention mechanism. Temporal distribution of attention in Grid $_{6\times6}$ roadnet learned by MFRAD is demonstrated in Fig. 4. Similarly as traffic jam, flow of Inter I0 changes greatly that flow from Inter I4 to Inter I0 decreases while increases from Inter I3 to Inter I0. As shown in Fig. 4(b) the score of SELF-attention occupies the largest blue area while that of I4 and I2 decreases and I3 and I1 increases, indicating that the attention scores match with the real traffic condition. Thus, MFRAD is verified to capable of approximating accurate weighted mean-field neighbor effect.



(a) (b)

Fig. 4. Temporal distribution of attention score when facing with changeable traffic flow.(a) Roadnet of Inter I0. (b) Temporal distribution of attention score of Inter I0.

6 Conclusion

In this paper, we develop a weighted mean-field multi-agent reinforcement learning algorithm via reward attribution decomposition, which incorporates weighted information of neighbors with attention mechanism to capture the dynamic influence of others. Considering the negative effect of hysteresis information, reward attribution decomposition is integrated to decompose the pairwise mean-field Q-function as SELF-term and neighbor term which represents the local effect and mean-field effect of neighbors respectively, realizing the fully decentralized execution without any information exchanging. Experiments in various many-agent scenarios demonstrate that MFRAD boosts the performance and stabilizes the training process and also has great scalability to real-world tasks.

Acknowledgment. This work was supported in part by the National Key Research and Development Program of China (No. 2020AAA0107400), STCSM (No. 18DZ2271000 and 19ZR141420), NSFC (No. 12071145) and the Fundamental Research Funds for the Central Universities.

References

1. Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., Li, Z.: Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. AAAI **34**(04), 3414–3421 (2020)
2. Fang, B., Wu, B., Wang, Z., Wang, H.: Large-scale multi-agent reinforcement learning based on weighted mean field. In: ICCSSP. pp. 309–316. Springer (2020)
3. Ganapathi Subramanian, S., Poupart, P., Taylor, M.E., Hegde, N.: Multi type mean field reinforcement learning. In: AAMAS (2020)
4. Ganapathi Subramanian, S., Taylor, M.E., Crowley, M., Poupart, P.: Partially observable mean field reinforcement learning. In: AAMAS (2021)
5. Guo, X., Hu, A., Xu, R., Zhang, J.: Learning mean-field games. In: NeurIPS (2019)
6. Gupta, J.K., Egorov, M., Kochenderfer, M.: Cooperative multi-agent control using deep reinforcement learning. In: AAMAS (2017)
7. Jeong, S.H., Kang, A.R., Kim, H.K.: Analysis of game bot's behavioral characteristics in social interaction networks of mmorpg. ACM SIGCOMM Computer Communication Review **45**(4), 99–100 (2015)
8. Jiang, J., Dun, C., Huang, T., Lu, Z.: Graph convolutional reinforcement learning. In: ICLR (2020)
9. Li, M., Qin, Z., Jiao, Y., Yang, Y., Wang, J., Wang, C., Wu, G., Ye, J.: Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In: WWW (2019)
10. Li, W., Wang, X., Jin, B., Sheng, J., Hua, Y., Zha, H.: Structured diversification emergence via reinforced organization control and hierarchical consensus learning. In: AAMAS (2021)
11. Li, W., Wang, X., Jin, B., Sheng, J., Zha, H.: Dealing with non-stationarity in MARL via trust region decomposition. In: ICLR (2022)
12. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: NeurIPS (2017)

13. Mao, H., Liu, W., Hao, J., Luo, J., Li, D., Zhang, Z., Wang, J., Xiao, Z.: Neighborhood cognition consistent multi-agent reinforcement learning. In: AAAI (2020)
14. Mao, H., Zhang, Z., Xiao, Z., Gong, Z., Ni, Y.: Learning multi-agent communication with double attentional deep reinforcement learning. AAMAS **34**(1), 1–34 (2020)
15. Matignon, L., Laurent, G.j., Le fort piat, N.: Review: Independent reinforcement learners in cooperative markov games: A survey regarding coordination problems. Knowl. Eng. Rev. **27**(1), 1–31 (2012)
16. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
17. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S.: Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: ICML (2018)
18. Ren, W.: Represented value function approach for large scale multi agent reinforcement learning. Arxiv (2020)
19. Sheng, J., Hu, Y., Zhou, W., Zhu, L., Jin, B., Wang, J., Wang, X.: Learning to schedule multi-numa virtual machines via reinforcement learning. Pattern Recognition **121**, 108254 (2022)
20. Sheng, J., Wang, X., Jin, B., Yan, J., Li, W., Chang, T.H., Wang, J., Zha, H.: Learning structured communication for MARL. ArXiv (2020)
21. 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: ICML (2019)
22. 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: AAMAS (2018)
23. Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: ICML (1993)
24. Varaiya, P.: Max pressure control of a network of signalized intersections. Transportation Research Part C: Emerging Technologies **36**, 177–195 (2013)
25. Yang, F., Vereshchaka, A., Chen, C., Dong, W.: Bayesian multi-type mean field multi-agent imitation learning. In: NeurIPS (2020)
26. Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., Wang, J.: Mean field multi-agent reinforcement learning. In: ICML (2018)
27. Ye, D., Chen, G., Zhang, W., Chen, S., Yuan, B., Liu, B., Chen, J., Liu, Z., Qiu, F., Yu, H., Yin, Y., Shi, B., Wang, L., Shi, T., Fu, Q., Yang, W., Huang, L., Liu, W.: Towards playing full MOBA games with deep reinforcement learning. In: NeurIPS (2020)
28. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: CVPR (2017)
29. Zhang, T., Xu, H., Wang, X., Wu, Y., Keutzer, K., Gonzalez, J.E., Tian, Y.: Multi-agent collaboration via reward attribution decomposition. Arxiv (2020)
30. Zheng, L., Yang, J., Cai, H., Zhou, M., Zhang, W., Wang, J., Yu, Y.: Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In: AAAI (2018)
31. Zhou, M., Jin, J., Zhang, W., Qin, Z., Jiao, Y., Wang, C., Wu, G., Yu, Y., Ye, J.: Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. In: CIKM. pp. 2645–2653 (2019)
32. Zimmer, M., Glanois, C., Siddique, U., Weng, P.: Learning fair policies in decentralized cooperative multi-agent reinforcement learning. In: ICML (2021)