MFVFD : A Multi-Agent Q-Learning Approach to Cooperative and Non-Cooperative Tasks

Tianhao Zhang[*1] , **Qiwei Ye**[*2] , **Jiang Bian**[2] , **Guangming Xie**[1] and **Tie-Yan Liu**[2]
[1]Peking University
[2]Microsoft Research Asia
{tianhao_z, xiegming}@pku.edu.cn, {qiwye, jiabia, tyliu}@microsoft.com

Abstract

Value function decomposition (VFD) methods under the popular paradigm of centralized training and decentralized execution (CTDE) have promoted multi-agent reinforcement learning progress. However, existing VFD methods proceed from a group's value function decomposition to only solve cooperative tasks. With the individual value function decomposition, we propose MFVFD , a novel multi-agent Q-learning approach for solving cooperative and non-cooperative tasks based on mean-field theory. Our analysis on the 'Hawk-Dove' and 'Nonmonotonic Cooperation' matrix games evaluate MFVFD 's convergent solution. Empirical studies on the challenging mixed cooperative-competitive tasks where hundreds of agents coexist demonstrate that MFVFD significantly outperforms existing baselines.

1 Introduction

Decision-making towards global optimization under a complex and non-stationary multi-agent environment requires each intelligent agent to perceive their environments as well as the interactions between other agents. Multi-agent deep reinforcement learning (MARL) holds considerable promise to help address a variety of real-world problems, either in a competitive setting, such as coordinating self-driving vehicles [Shalev-Shwartz *et al.*, 2016] or traffic signals in a transportation system [Calvo and Dusparic, 2018] and investing in financial markets [Schmid *et al.*, 2018], or in a cooperative setting, such as optimizing the utility of a smart grid [Dimeas and Hatziargyriou, 2010] or Internet of Things (IoT) [Deng *et al.*, 2020].

However, pursuing effective MARL yields two major challenges: partial observability and scalability limitations [Buşoniu *et al.*, 2010]. Particularly, the partial observability of the entire environment, though increasing the efficiency of decision-making, may severely limit each agent's ability to find its optimal actions. On the other hand, optimizing using all information on the environment may lead to a huge joint state-action space, which exponentially grows with the number of agents, causing scalability limitations. Many recent research advances [Lowe *et al.*, 2017; Sunehag *et al.*, 2018; Rashid *et al.*, 2018; Son *et al.*, 2019] have attempted to address these two challenges from two directions. In one of the directions, researchers have sought to address the partial observability issue by proposing a centralized training with decentralized execution (CTDE) MARL paradigm [Lowe *et al.*, 2017], where agents' policies are trained with access to global information in a centralized manner and executed only based on local observations in a decentralized manner. To further resolve the scalability limitation by CTDE, certain value function decomposition algorithms based on the IGM (Individual-Global-Max) principle are proposed, including VDN [Sunehag *et al.*, 2018], QMIX [Rashid *et al.*, 2018] and QTRAN [Son *et al.*, 2019], where the IGM emphasizes that optimal joint action should be equivalent to the collection of individual optimal actions of agents. However, these value function decomposition methods aim to factorize the value function of the team to the collection of value functions of team members, which limits their scope merely in cooperative MARL. That is, none of them can be applied to widely existing non-cooperative environments.

Another major approach takes advantage of the mean-field theory to solve the scalability issue in stochastic games, where the mean-field theory considers that the interactions within the population of agents can be approximated by that of a single agent played with the average effect from the population [Domb, 2000; Lasry and Lions, 2007; Huang *et al.*, 2006]. That is, the joint state-action can be replaced by the state-action distribution to reduce the dimension of the joint state-action space. Yang [2018] uses the mean action of neighbors as the empirical distribution of the actions to approximate the joint action. However, it is only applicable when all agents are of the same type under its assumption. Using K-means algorithm, Subramanian [2020] extends it to multiple types by approximating the joint action of N agents to N mean actions. Unfortunately, this method could fail with an increasing total number of agent classes. In addition, neither of the two methods carry out decentralized execution; their strategies rely on the acquisition or estimation of the actions of neighbors, which work little in scenarios where communication or observation is limited.

In this paper, we aim to tackle MARL to have it scalable when many agents coexist under partial observability limita-

*Correspondence to: Tianhao Zhang, Qiwei Ye

tions and have it applicable to cooperative, competitive, or mixed tasks. To this end, we propose a new approach by combining the advantages of **M**ean-**F**ield theory and **V**alue **F**unction **D**ecomposition, called MFVFD . MFVFD introduces that each agent's own efforts are affected by the population's average effect. Thus, the standard individual Q-function (based on global state and joint actions) of each agent can be transformed into the sum of its local Q-function (based on its local observation and action) and its mean-field Q-function (based on its neighbors' both observation and action distribution). This way, MFVFD pioneers the decomposition of the joint Q-function from the individual's perspective rather than that of the team like previous value decomposition works. Thus, not only can it be applied to cooperation tasks but also non-cooperation tasks. To our best knowledge, MFVFD is the first multi-agent Q-learning algorithm that effectively achieves high scalability at the individual level under the partial observation constraint.

We assessed the performance of MFVFD by comparing it against state-of-art MARL algorithms in four environments. First, we consider different types of single-state matrix games, including the *Hawk-Dove* non-cooperation matrix game and *Nonmonotonic Cooperation Matrix Game*. Results show that our proposed approach converges to the pure Nash Equilibrium (NE) in non-cooperation game and successfully finds the Pareto Optimal solution in the cooperative game. We then observed its cooperation ability in the Cooperative Navigation environment and further evaluated its performance in a more challenging Mixed-Cooperation-Competition game with 400 agents, MAgent [Zheng *et al.*, 2017]. Empirical results show that MFVFD significantly outperforms other multi-agent baselines. To further understand the efficacy of MFVFD , we evaluated MFVFD on a range of tasks on FLOW, a traffic control benchmark [Wu *et al.*, 2017], and the results show that MFVFD can converge faster than the baseline with a better final performance.

2 Background

2.1 Stochastic Game

We took the stochastic game (SG) [Littman, 1994] as the standard for modeling discrete-time and non-cooperative N-agent multi-agent tasks. As in many previous work, an SG can be defined as a tuple $< \mathcal{S}, \mathcal{A}^1, \ldots, \mathcal{A}^N, r^1, \ldots, r^N, p, \gamma >$, where $s \in \mathcal{S}$ denotes the true state of the environment. Each agent $i \in \mathcal{N} := \{1, \ldots, N\}$ chooses an action $a^i \in \mathcal{A}^i$ at each time step. The reward function for agent i is defined as $r^i : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \mathbb{R}$, which determines the immediate reward. $p : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \Omega(\mathcal{S})$ characterizes all state transition dynamics. The constant $\gamma \in [0, 1)$ is the discount factor. When the state is not fully observed, the system is called a *partially observed Stochastic Game*, where each agent i has individual observation $o^i \in \mathcal{O}^i$, which is typically some function of the state s.

An SG can be considered a sequence of normal-form games, which are games that can be represented in a matrix [Yang and Wang, 2020]. Take the *Hawk-dove* game [Grafen, 1979] as an example (shown in Section 4.1). Agents can be either aggressive like a hawk (action 'A') or timid like a dove (action 'B'). In this scenario, if both agents care only about maximizing their own expected reward without considering other agents (the solution concept in a single-agent RL), both agents reaching the action of hawks resulting in the destructive outcome. If both agents choose to cooperate with others, acting like doves, then the outcome is peaceful. However, one acting like a hawk can take advantage of the other acting like a dove, which will break this 'peaceful' solution. That is to say, strategies that only consider self-interest will be ruined, and strategies that only consider the team will be used.

2.2 Solving Stochastic Games

Nash equilibrium [Nash, 1951] is the baseline solution concept for the stochastic game, which denotes the steady-state where none of the agents will have a motivation to deviate from their best response give others. The NE of the *Hawk-dove* game are (hawk, dove) and (dove, hawk), representing that the best response is when one agent acts like a dove and the other acts like a hawk. Nash Q-learning [Hu and Wellman, 2003] introduced NE into Q-learning for solving stochastic games, which estimate the Q-function according to the NE value. However, like some traditional equilibrium-solving approaches [Bowling and Veloso, 2002], Nash Q-learning has the problem of high computational complexity, so it cannot be applied to scenarios where multiple agents coexists.

Mean-field theory [Domb, 2000], concerning the approximation of interactions between agents into the average effect from the overall population, is a solution for solving scalability to SG. The marriage of the mean-field theory and multi-agent reinforcement learning gives to the framework of mean-field reinforcement learning (MFRL), which has recently attracted widespread attention. Yang [2018] approximated the standard Q-function using the mean action of the neighbor agents, which reduces the computational complexity of the joint action. In their work, the joint Q function for each agent is decomposed into local Q functions that capture pairwise interactions:

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

where N^i is the number of neighbors of the agent i and $\mathcal{N}(i)$ is the index set of neighboring agents. Then, the joint Q function is approximated by the mean field Q-function $Q^i(s, \boldsymbol{a}) \approx Q^i_{\text{MF}}(s, a^i, \bar{a}^i)$ under its assumptions. The mean action $\bar{a}^i = \frac{1}{N^i} \sum_{k \in \mathcal{N}(i)} a^k$ represents the neighbor action distribution, where a^k is the action of each neighbor k.

However, this method has three limitations: 1) all agents should be homogeneous (same type), 2) insufficient expression ability, 3) unable to distributed execute with local information. Subramanian [2020] used clustering approaches to approximating M types of agents for solving the first limitation, where $Q^i(s, \boldsymbol{a}) \approx Q^i_{\text{MTMF}}(s, a^i, \bar{a}^i_1, ..., \bar{a}^i_M)$. Unfortunately, the computation complexity of this method increases as the types of agents increase. Besides, these two methods cannot accurately estimate Q-function in some cases, such as $[Q^i(s, 0, 1, 1) = 6, Q^i(s, 0, 2, 2) = 0, Q^i(s, 0, 1, 2) = 2]$. Based on $Q^i(s, 0, 1, 1) = \frac{1}{2} Q^i(s, 0, 1) + \frac{1}{2} Q^i(s, 0, 1) =$

6, $Q^i(s,0,1) = 6$. Similarly, based on $Q^i(s,0,2,2) = \frac{1}{2}Q^i(s,0,2) + \frac{1}{2}Q^i(s,0,2) = 0$, $Q^i(s,0,2) = 0$. Then, $Q^i(s,0,1,2) = \frac{1}{2}Q^i(s,0,1) + \frac{1}{2}Q^i(s,0,2) = 3$. However, $Q^i(s,0,1,2)$ should be 2. Thus, the oscillate will occur during the learning process. In addition, they cannot carry out decentralized execution from local exploration due to dependence on the global state and action observation.

2.3 CTDE and VFD

Centralized training with decentralized execution (CTDE) is a popular paradigm of MARL tasks. Through centralized training, the action-observation of all agents and the full state can be made accessible to all agents. In this manner, agents can learn and construct individual action-value functions correctly while selecting actions based on their own local observation at the execution time without having to refer to the joint one, which can solve the issue of partial observability in some cases. However, during centralized training, CTDE requires a joint action-value function $Q(\boldsymbol{s}, \boldsymbol{a})$ conditioned on the global state and join action, which is difficult to learn when there are many agents.

Value function decomposition (VFD) [Koller and Parr, 1999; Guestrin *et al.*, 2001; Sunehag *et al.*, 2018; Rashid *et al.*, 2018; Son *et al.*, 2019] methods have been proposed to handle a joint action-value function under the CTDE paradigm, where VDN [Sunehag *et al.*, 2018] has received a great amount of attention and seen widespread application with its simple ideas, described as $Q_{jt}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=1}^{N} Q_i(\tau_i, a_i)$, where τ represents an action-observation history. QMIX [Sunehag *et al.*, 2018], QTRAN [Son *et al.*, 2019], and QPLEX [Wang *et al.*, 2020] improve the expressive power of decomposition to perfect VDN. Although these VFD methods can ensure the consistency of the optimal joint action and the local optimal actions, all they factorize the team Q-function with the shared reward into individual Q-functions, which restricts them from solving SG.

3 MFVFD : Learn to Factorize with Mean-Field

In this section, we will introduce the MFVFD , a novel factorization solution taking advantage of mean-field theory for multi-agent systems. The main idea behind our multi-agent learning approach is to factorize the original individual joint Q-function of each agent by considering the influence of the neighbors. Figure 1 illustrates the main components of our approach.

3.1 Factorization Approximation

As a generally simplified approach illustrated in Equation (1), the global interactions can be factorized as the pairwise local interactions between any pair of agents implicitly [Blume and others, 1993], where the weight of each pairwise local interaction is equal ($\frac{1}{N^i}$). Considering that different states and actions may have different effects, we believe that each pairwise local interaction's weight may differ. Thus, the pairwise local interactions should satisfy

$$Q^i(s, \boldsymbol{a}) = \sum_{k \in \mathcal{N}(i)} \lambda^i(o^i, o^k, a^i, a^k) Q^i(s, a^i, a^k), \quad (2)$$

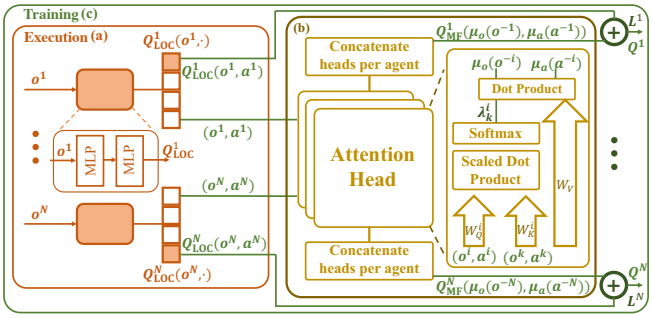


Figure 1: MFVFD Architecture. (a) In the execution, each agent chooses an action based on the individual network Q^i_{LOC} with its local observation o^i; (b) then, all the observation-action pairwise $[(o^i, a^i)]_{i=1}^N$ are passed into a shared mean-field network with multiple heads attention to calculate the mean-field Q-function for each agent Q^i_{MF}; (c) lastly, based on the $Q^i := Q^i_{\text{LOC}} + Q^i_{\text{MF}}$, each agent's strategy is updated in the training.

where $\lambda^i(o^i, o^k, a^i, a^k) \in [0, 1]$, which is the weight function representing the correlation of agent k for agent i. Thus, $\sum_{k \in \mathcal{N}(i)} \lambda^i(o^i, o^k, a^i, a^k) = 1$. In this manner, when $\lambda^i(o^i, o^1, 0, 1) = \frac{1}{3}$, the Q-functions in the above example (Section 2.2) can be accurately estimated.

Then, under the CTDE paradigm, Equation (2) can be further decomposed:

$$\begin{aligned} Q^i(s, \boldsymbol{a}) &= \sum_k \lambda^i(o^i, o^k, a^i, a^k) Q^i(s, a^i, a^k) \\ &= \sum_k \lambda^i_k \cdot \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r^i(s_t, a^i_t, a^k_t) | s_1 = s, a_1 = \mathbf{a}] \\ &= \sum_k \lambda^i_k \cdot \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} (r^i(o^i_t, a^i_t) + r^i(o^k_t, a^k_t)) | s_1 = s, a_1 = \mathbf{a}] \\ &:= \bar{Q}^i_{\text{i}}(s, \boldsymbol{a}) + \sum_k \lambda^i_k \cdot \bar{Q}^i_{\text{k}}(s, \boldsymbol{a}), \end{aligned} \quad (3)$$

where λ^i_k is short for $\lambda^i(o^i, o^k, a^i, a^k)$, $\bar{Q}^i_{\text{i}}(s, \boldsymbol{a}) := \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r^i(o^i, a^i) | s_1 = s, a_1 = \boldsymbol{a}]$, $\bar{Q}^i_{\text{k}}(s, \boldsymbol{a}) := \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r^i(o^k_t, a^k_t) | s_1 = s, a_1 = \boldsymbol{a}]$. The key insight of this decomposition is the reward $r^i(s, a^i, a^k)$ of agent i from the pairwise local interaction with agent k can be considered as the sum of the agent i's individual effort $r^i(o^i, a^i)$ and the influence $r^i(o^k, a^k)$ to agent i caused by agent k, i.e., $r^i(s, a^i, a^k) = r^i(o^i, a^i) + r^i(o^k, a^k)$. Note that $r^i(o^i, a^i)$ and $r^i(o^k, a^k)$ are dummy values for auxiliary factorization. If the local observation and action are sufficient to model $\bar{Q}^i_{\text{i}}(s, \boldsymbol{a})$ and $\bar{Q}^i_{\text{k}}(s, \boldsymbol{a})$, we could expect the following approximation to be valid:

$$\begin{aligned} Q^i(s, \boldsymbol{a}) &:= \bar{Q}^i_{\text{i}}(s, \boldsymbol{a}) + \sum_k \lambda^i_k \cdot \bar{Q}^i_{\text{k}}(s, \boldsymbol{a}) \\ &\approx Q^i_{\text{LOC}}(o^i, a^i) + \sum_{k \in \mathcal{N}(i)} \lambda^i_k Q^i_{\text{NB}}(o^k, a^k), \end{aligned} \quad (4)$$

where agent i's local Q-function $Q^i_{\text{LOC}}(o^i, a^i)$ represents its own efforts conditioned on local information, and its neigh-

bor Q-function Q^i_{NB} represents the influence to agent i caused by its neighbors.

3.2 Mean Field Approximation

We utilize the mean-field theory [Domb, 2000] to approximate $\sum_k \lambda^i(o^i, o^k, a^i, a^k)Q^i_{\text{NB}}(o^k, a^k)$ for further solving scalability. On the basis of the previous MFRL approaches (described in Section 2.2) using the mean action $\bar{a}^i = \frac{1}{N}\sum_k a^k$ to represent agent i's neighborhood action distribution, we calculate the *weighted average* to generalize it. As for agent i, each neighbor k's action (resp., observation) can be calculated as the sum of the neighborhood action distribution $\mu_a(\boldsymbol{a}^{-i})$ (resp., neighborhood observation distribution $\mu_o(\boldsymbol{o}^{-i})$) and a small fluctuation δa^i_k (resp., δa^i_k):

$$\begin{aligned} o^k &= \mu_o(\boldsymbol{o}^{-i}) + \delta o^i_k, \quad where \quad \mu_o(\boldsymbol{o}^{-i}) = \sum_{k\in\mathcal{N}(i)} \lambda^i_k o^k \\ a^k &= \mu_a(\boldsymbol{a}^{-i}) + \delta a^i_k, \quad where \quad \mu_a(\boldsymbol{a}^{-i}) = \sum_{k\in\mathcal{N}(i)} \lambda^i_k a^k. \end{aligned} \tag{5}$$

By Taylor's theorem, the neighbor Q-function $Q^i_{\text{NB}}(o^k, a^k)$, if twice-differentiable *w.r.t* the action a^k and observation o^k taken by neighbor k, can be expanded and expressed as:

$$\begin{aligned} &Q^i(s, \boldsymbol{a}) \approx Q^i_{\text{LOC}}(o^i, a^i) + \sum_k \lambda^i_k Q^i_{\text{NB}}(o^k, a^k) \\ &= Q^i_{\text{LOC}}(o^i, a^i) + \sum_k \lambda^i_k \big[Q^i_{\text{NB}}(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i})) \\ &\quad + (\delta o^i_k \bigtriangledown_{\mu_o} + \delta a^i_k \bigtriangledown_{\mu_a}) Q^i_{\text{NB}}(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i}))) + R_2\big] \\ &= Q^i_{\text{LOC}}(o^i, a^i) + Q^i_{\text{NB}}(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i})) + R_2, \end{aligned} \tag{6}$$

where the first-order term of the Taylor extension is dropped since the $\sum_k \lambda^i_k \delta o^i_k = 0$ and the $\sum_k \lambda^i_k \delta a^i_k = 0$. In addition, the R_2, which is the Taylor polynomial's remainder, can be seen as a small fluctuation (the proof refers to [Yang *et al.*, 2018]). Considering that $\mu_a(\boldsymbol{a}^{-i}) = \sum_k \lambda^i_k o^k = N \times \boldsymbol{\lambda} \cdot \bar{\boldsymbol{a}}^{-i}$, where $\bar{\boldsymbol{a}}^{-i} = \sum_k \frac{1}{N} a^k$ can be seen as the empirical distribution of neighborhood's action [Yang *et al.*, 2018; Subramanian *et al.*, 2020], Q^i_{NB} conditioned on μ_o and μ_a can be regarded as the mean field Q-function of agent i. Therefore, Equation (6) can be remarked as:

$$Q^i(s, \boldsymbol{a}) \approx Q^i_{\text{LOC}}(o^i, a^i) + Q^i_{\text{MF}}(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i})). \tag{7}$$

3.3 The MFVFD Architecture

This section introduces the architecture of MFVFD which is illustrated in Figure 1. The overall architecture of our proposed method consists of two parts: *local part* and *mean-field part*. The local part includes *Local Action-Value Function* Q^i_{LOC} for each agent i, which has the related local action-value network with parameters α_i. The network takes agent i's own observation and action (o^i, a^i) as input and produces local Q value $Q^i_{\text{LOC}}(o^i, a^i)$.

As for the mean-field part, we apply the attention method with the 'query-key' mechanism [Vaswani *et al.*, 2017] to construct the observation and action distribution (μ_o and μ_a). Specifically, for each agent i, we denote the local observation-action (o^i, a^i) as X^i, then, X^i can be transformed into a 'query' by a matrix W_Q. Similarly, the neighbor k's local information X^k can be transformed into a 'key' by a matrix W_K. Then, the query and the key is embedded into a *Softmax* function to calculate the similarity between these two as λ^i_k:

$$\lambda^i_k = \frac{exp(X^{k\intercal} W_K{}^\intercal W_Q X^i)}{\sum_j exp(X^{j\intercal} W_K{}^\intercal W_Q X^i)}, \tag{8}$$

which can be learned by a network with parameters ω^i, where W_Q and W_K are the networks parameters. The multi-heads approach can be utilized for considering the correlation from different angles, where each head corresponds to a separate set of parameters (W_K, W_Q). In this way, the μ_o and μ_a can be described by:

$$\big(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i})\big) = \frac{1}{M}\sum_m^M \sum_k \lambda^i_{\omega_m, k} X^k, \tag{9}$$

where M is the number of the head. Thus, the mean-field part for each agent i includes *Weight Vector Functions* $[\lambda^i_m]^M_{m=1}$ and *Mean Field Value Function* Q^i_{MF}, which has two related networks:

(i) weight vector network with parameters ω^i takes local observation and action pairwise (o^i, o^k, a^i, a^k) as input and produces the credit $\lambda^i(o^i, o^k, a^i, a^k)$ to calculate μ_o and μ_a.

(ii) mean field network with parameters β_i takes $(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i}))$ as input and produces the mean field value for each agent i as $Q^i_{\text{MF}}(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i}))$.

In this manner, the structure of MFVFD can be described as:

$$Q^i(s, \boldsymbol{a}) \approx Q^{i,\alpha_i}_{\text{LOC}}(o^i, a^i) + Q^{i,\beta_i}_{\text{MF}}(\mu_o(\boldsymbol{o}^{-i}), \mu_a(\boldsymbol{a}^{-i})). \tag{10}$$

The pseudocode of our method is shown in Algorithm 1. All the networks are trained in the centralized training manner, where the individual Q-function Q^i is trained to minimize:

$$\begin{aligned} J_{Q^i}(\alpha_i, \beta_i, [\omega_m]^M_{m=1}) = \mathbb{E}_{\boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{o}_{t+1}, \boldsymbol{a}_{t+1}} \Bigg[\Bigg(\big(Q^i_{\text{LOC}}(o^i, a^i) \\ + Q^i_{\text{MF}}(\mu_o(o^{-i}_t), \mu_a(a^{-i}_t))\big) - y^i \Bigg)^2 \Bigg], \end{aligned} \tag{11}$$

where y^i is as:

$$y^i = r^i_t + \gamma\big(\hat{Q}^i_{\text{LOC}}(o^i_{t+1}, a^i_{t+1}) + \hat{Q}^i_{\text{MF}}(\mu_o(o^{-i}_{t+1}), \mu_a(a^{-i}_{t+1}))\big). \tag{12}$$

The $\hat{Q}_{\text{LOC}}$ and $\hat{Q}_{\text{MF}}$ come from the target network with parameters $\hat{\alpha}^i$ and $\hat{\beta}^i$, respectively, which has been shown to stabilize training [Van Hasselt *et al.*, 2016]. Besides, to guarantee $\arg\max_{a^i} Q^i = \arg\max_{a^i} Q^i_{\text{LOC}}$, we define the local optimal actions of neighbors as $\boldsymbol{a}^{-i,*}$, which is calculated based on the $[\arg\max_{a^k} Q^k_{\text{IND}}]_{k\in\mathcal{N}(i)}$. We also define the local optimal action of agent i based on Q^i_{IND} is $a^{i,*}$. Then, the mean-field function is constraint by:

$$\begin{aligned} J_{Q^i_{\text{MF}}}(\beta) = \mathbb{E}_{s\sim D} \Bigg[\Bigg(\hat{Q}^i_{\text{MF}}(\mu_o, \mu_a) | \boldsymbol{a}^{-i,*}, a^{i,*} \\ - \max_{a^{i,'}} Q^i_{\text{MF}}(\mu_o, \mu_a) | \boldsymbol{a}^{-i,*}, a^{i,'} \Bigg)^2 \Bigg] \end{aligned} \tag{13}$$

Algorithm 1 Mean field value decomposition

Initial $Q_{\text{LOC}}^{\alpha_i}, \hat{Q}_{\text{LOC}}^{\hat{\alpha}_i}, Q_{\text{MF}}^{\beta_i}, \hat{Q}_{\text{MF}}^{\hat{\beta}_i}$ for all agents.
while training not finished **do**
  for each agent i **do**
    Sample action a^i from $Q_{\text{LOC}}^{\alpha_i}$ with ϵ-greedy policy.
  end for
  Take joint observations $\boldsymbol{o} = [o^i]_{i=1}^N$, joint actions $\boldsymbol{a} = [a^i]_i^N$, joint reward $\boldsymbol{r} = [r^i]_i^N$ and joint next observations $\boldsymbol{o'} = [o'^i]_i^N$.
  Store $< \boldsymbol{o}, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{o'} >$ in the replay buffer D.
  Sample a mini-batch of K experiences from D.
  Get next actions $\boldsymbol{a'} = [a'^i]_i^N$ from $[\hat{Q}_{\text{LOC}}^{\hat{\alpha}_i}]_i^N$.
  for each agent i **do**
    Set y^i based on Equation (12).
    Update the Q-networks based on Equation (11) and Equation (13).
  end for
  Update the parameters of the target network for each agent i with learning rate τ:
  $\alpha_i' \leftarrow \tau\alpha_i + (1-\tau)\alpha_i'$
  $\beta_i' \leftarrow \tau\beta + (1-\tau)\beta_i'$
end while

4 Experiment

This section will first consider matrix games, including the non-cooperative and cooperative types, to investigate the convergent solution of MFVFD . Then, we will evaluate the performance of MFVFD on the widely utilized testbeds including the Cooperative Navigation [Mordatch and Abbeel, 2017] and the Mixed Cooperative-Competitive MAgent [Zheng *et al.*, 2017]. Last, we chose the traffic environment Flow [Wu *et al.*, 2017] to show that MFVFD has the potential to solve complex real-world problems. The structure of Q_{LOC}^i, Q_{MF}^i in practice are simple fully connected networks with 2 hidden layers, where each layer has 64 neurons with *ReLU* activation.

4.1 Matrix Games

To ensure sufficient data collection in the joint action space, we adopted the $\epsilon-$greedy for $50k$ steps.

Non-cooperative matrix game. We chose the 'Hawk-dove' matrix game, whose payoff matrix is shown in Table 1a. Table 1c shows that MFVFD converges to one of the NE, (dove, hawk). We repeated this experiment 50 times, of which 24 times it converged to (hawk, dove), and 26 times it converged to (dove, hawk). We compared it with the single-RL method, DQN, and the VFD method, VDN. DQN with the local information oscillated from (Hawk, Hawk) to (Hawk, dove) or (dove, hawk) during training process. Because the 'hawk' action with the maximum return and the maximum average return will produce destructive reward when all agents chose it. DQN, which tends to maximize agent's own reward without considering the actions of other agents naturally oscillates. The convergent solution of VDN is (dove, dove), which is the maximum return of the team. However, in SG, such a cooperative strategy can be easily used by selfish agents, then the payoff will reduce from 5 to 1.

a_1 \ a_2	A	B
A	0,0	**8,1**
B	**1,8**	5,5

(a) Payoff of *Hawk-dove*

a_1 \ a_2	A	B
A	0.6,-5.6	8.6,-1.6
B	0.7, 2.3	4.8,2.3

(b) MFVFD :$Q_{\text{mf}}^1, Q_{\text{mf}}^2$

	A	B
Q_{LOC}^1	-0.6	**0.3**
Q_{LOC}^2	**5.7**	2.7

(c) MFVFD :$Q_{\text{LOC}}^1, Q_{\text{LOC}}^2$

Q^1 \ Q^2	A	B
A	0.0,0.1	8.0,1.1
B	**1.0,8.0**	5.1,5.0

(d) MFVFD :Q^1, Q^2

Table 1: Payoff matrix of *Hawk-dove* matrix game and reconstructed Q results on the game, where action 'A' represents 'Hawk', 'B' represents 'Dove'. Boldface means optimal/greedy actions from the state-action value, where (A,B) and (B,A) are pure NE.

a_1 \ a_2	A	B	C
A	8	-12	-12
B	-12	0	0
C	-12	0	0

(a) Payoff of *cooperative game*

a_1 \ a_2	A	B	C
A	3.2	-12.2	-12.1
B	-13.6	3.0	3.1
C	-13.7	2.9	3.0

(b) MFVFD :$Q_{\text{mf}}^1, Q_{\text{mf}}^2$

Q_{LOC}^1 \ Q_{LOC}^2	3.1	-1.5	-1.6
1.7	4.8	0.2	0.1
-1.5	1.6	-3.0	-3.1
-1.4	1.7	-2.9	-3.0

(c) MFVFD :$Q_{\text{LOC}}^1, Q_{\text{LOC}}^2$

a_1 \ a_2	A	B	C
A	8.0	-12.0	-12.0
B	-12.0	0.0	0.0
C	-12.0	0.0	0.0

(d) MFVFD :Q^1, Q^2

Table 2: Payoff matrix of Non-Monotonic cooperation game and reconstructed Q results on the game. Boldface means optimal/greedy actions from the state-action value, where (A,A) is the joint optima.

Cooperative matrix game. The cooperative game is a special case of the SG when the reward of each agent r^i is equal. We chose the *Non-Monotonic Cooperation Matrix* [Son *et al.*, 2019] for evaluating the convergent solution in the cooperative matrix game. As shown in Table 2c, with the assistance of the mean-field value (Table 2b), MFVFD achieves the optimal joint actions, while QTRAN [Son *et al.*, 2019] showed that VDN and QMIX failed to learn this optimal joint actions. Details are shown in Supplementary files.

4.2 Cooperative Navigation

As shown in Figure 2(A), the *Cooperative Navigation* task makes agents cooperate through discrete actions to reach a set of L landmarks [Mordatch and Abbeel, 2017]. In addition, during the move, agents must not collide with each other or they will be punished. The SAC [Haarnoja *et al.*, 2017], MADDPG [Lowe *et al.*, 2017], VDN and QMIX are chosen as the baselines. Each algorithm repeats the experiment five times under the same settings. The mean episode rewards are shown in Figure 2(B). Results show that MFVFD outperforms the state-of-the-art baselines in both convergence speed and value, which further illustrates our approach's extensibility on the cooperative task. See the supplementary material for details and related animations.

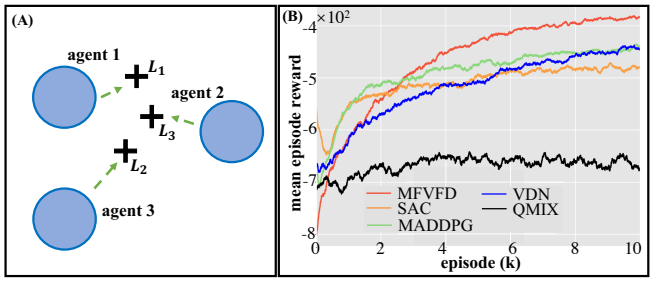


Figure 2: (A) Cooperative Navigation scenario. (B) MFVFD (red) outperforms baselines (SAC, MADDPG, VDN, QMIX) in both convergence speed and final performance.

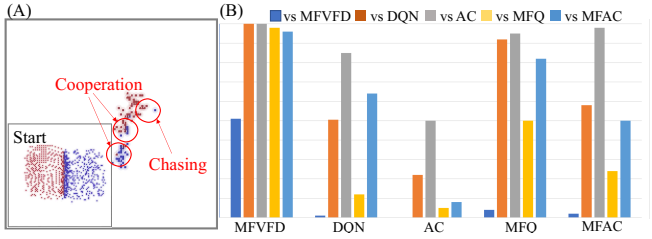


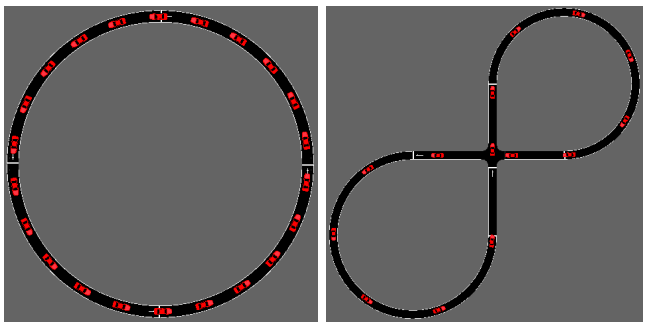
Figure 3: (A) In the mixed cooperative-competitive Battle game, there are two armies each with 400 agents, each agent should learn to destroy the enemy by cooperating with teammates and competing with opponents. (B) After self-play training, MFVFD basically completely defeated other baselines.

4.3 Mixed Cooperative-Competitive Game

Next, we chose the *Battle* task of *MAgent* [Zheng *et al.*, 2017] to further demonstrate the performance of MFVFD in the challenging mixed cooperative-competitive game with hundreds of agents. The *Battle* is a scenario with two armies fighting against each other in a grid world, where each army consists of 400 agents. Each agent has 23 valid actions to move, attack, and turn, and the goal of each army is to destroy the enemy and obtain more individual rewards. We chose DQN, Actor-Critic, MFQ and MFAC [Yang *et al.*, 2018], as our baselines. We trained all algorithms through self-play under the same settings. Then we let the different methods battle each other for 100 episodes, and the winning rates are shown in Figure 3(B). Noted that MFQ and MFAC using the global information during execution are always defeated by MFVFD with local information. See the supplementary material for details and related animations.

4.4 Traffic Control Benchmark

So far, we have introduced the excellent performance of MFVFD in the mixed cooperative-competitive task. In the real world, scenarios where cooperation and struggles coexist are everywhere. The traffic control benchmark is a typical example. As roads are limited, in order to drive fast, vehicles need to struggle with each other, but in order to avoid traffic jams, vehicles also need to cooperate. To demonstrate that MFVFD has the potential to solve real-world problems, we chose the traffic environment Flow [Wu *et al.*, 2017; Vinitsky *et al.*, 2018] to conduct exploratory experiments. The first scenario is *Ring* as shown in Figure 4(A) with 22



(A) Ring (B) Figure Eight

Figure 4: (A) The *Ring* environment has 22 vehicles where each aims to move quickly and avoid the rear-end collision on a one-way ring road. (B) The *Figure Eight* environment has 14 vehicles where each aims to move quickly and avoid rear-end collisions, side collisions, and traffic jams on a one-way eight road with a intersection.

episode	300	1000
MFVFD	28049.7	28623.0
VDN	18992.4	27705.8
DQN	13947.3	27131.2

(a) *Ring game*

episode	300	1000
MFVFD	4032.89	5174.25
VDN	3029.18	3419.2
DQN	2639.17	3126.69

(b) *Figure Eight game*

Table 3: Mean rewards of MFVFD and baselines in the traffic control benchmark tasks, where (a) shows *Ring* task, and (b) shows *Figure Eight* task.

RL controlled vehicles on a one-way ring road, which is the basis scene from the real world: drive as fast as possible on one-way streets while avoiding collisions by observing the front and rear distances. The second scenario is *Figure Eight*, which simulates the real problem of passing through a road intersection. We compared MFVFD with DQN and VDN under the same settings. The mean learning rewards are shown in Table 3. Experimental results show that MFVFD has the potential to solve practical traffic problems, and we will conduct more in-depth and comprehensive research in the future.

5 Conclusion & Future Work

In this paper, we introduced MFVFD , a novel multi-agent Q-learning framework under CTDE that leverages mean-field theory to decompose the joint action Q-function from the individual perspective, which further enables the scalability. Matrix game experiments evaluate the convergent solution of MFVFD . Empirical results show that MFVFD could achieve more significant performance comparing to previous methods in the mixed cooperative and competitive task with hundreds of agents.

In the near future, we aim to conduct additional experiments to compare across more complex tasks. While in the long term, it will be an interesting and valuable direction to study both efficacy and efficiency of MFVFD on real-world applications.

References

[Blume and others, 1993] Lawrence E Blume et al. The statistical mechanics of strategic interaction. *Games and economic behavior*, 5(3):387–424, 1993.

[Bowling and Veloso, 2002] Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[Buşoniu *et al.*, 2010] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pages 183–221, 2010.

[Calvo and Dusparic, 2018] Jeancarlo Arguello Calvo and Ivana Dusparic. Heterogeneous multi-agent deep reinforcement learning for traffic lights control. In *AICS*, pages 2–13, 2018.

[Deng *et al.*, 2020] Shuiguang Deng, Zhengzhe Xiang, Peng Zhao, Javid Taheri, Honghao Gao, Jianwei Yin, and Albert Y Zomaya. Dynamical resource allocation in edge for trustable internet-of-things systems: A reinforcement learning method. *IEEE Transactions on Industrial Informatics*, 16(9):6103–6113, 2020.

[Dimeas and Hatziargyriou, 2010] AL Dimeas and ND Hatziargyriou. Multi-agent reinforcement learning for microgrids. In *IEEE PES General Meeting*, pages 1–8. IEEE, 2010.

[Domb, 2000] Cyril Domb. *Phase transitions and critical phenomena*. Elsevier, 2000.

[Grafen, 1979] Alan Grafen. The hawk-dove game played between relatives. *Animal behaviour*, 27:905–907, 1979.

[Guestrin *et al.*, 2001] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored mdps. *Advances in neural information processing systems*, 14:1523–1530, 2001.

[Haarnoja *et al.*, 2017] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

[Hu and Wellman, 2003] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

[Huang *et al.*, 2006] Minyi Huang, Roland P Malhamé, Peter E Caines, et al. Large population stochastic dynamic games: closed-loop Mckean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information & Systems*, 6(3):221–252, 2006.

[Koller and Parr, 1999] Daphne Koller and Ronald Parr. Computing factored value functions for policies in structured mdps. In *IJCAI*, volume 99, pages 1332–1339, 1999.

[Lasry and Lions, 2007] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japanese journal of mathematics*, 2(1):229–260, 2007.

[Littman, 1994] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[Lowe *et al.*, 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

[Mordatch and Abbeel, 2017] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.

[Nash, 1951] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

[Schmid *et al.*, 2018] Kyrill Schmid, Lenz Belzner, Thomas Gabor, and Thomy Phan. Action markets in deep multi-agent reinforcement learning. In *International Conference on Artificial Neural Networks*, pages 240–249. Springer, 2018.

[Shalev-Shwartz *et al.*, 2016] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[Son *et al.*, 2019] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5887–5896. PMLR, 2019.

[Subramanian *et al.*, 2020] Sriram Ganapathi Subramanian, Pascal Poupart, Matthew E Taylor, and Nidhi Hegde. Multi type mean field reinforcement learning. *arXiv preprint arXiv:2002.02513*, 2020.

[Sunehag *et al.*, 2018] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-Decomposition Networks for Cooperative Multi-Agent Learning Based on team reward. In *AAMAS*, pages 2085–2087, 2018.

[Van Hasselt *et al.*, 2016] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Vinitsky *et al.*, 2018] Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M Bayen. Benchmarks for reinforcement learning in mixed-autonomy traffic. In *Conference on Robot Learning*, pages 399–409. PMLR, 2018.

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

[Wu *et al.*, 2017] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv preprint arXiv:1710.05465*, page 10, 2017.

[Yang and Wang, 2020] Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

[Yang *et al.*, 2018] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5571–5580. PMLR, 2018.

[Zheng *et al.*, 2017] Lianmin Zheng, Jiacheng Yang, Han Cai, Weinan Zhang, Jun Wang, and Yong Yu. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. *arXiv preprint arXiv:1712.00600*, 2017.