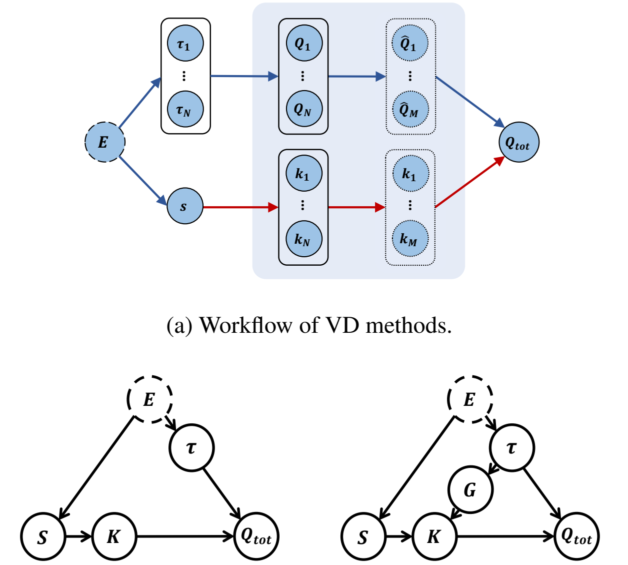
Deconfounded Value Decomposition for Multi-Agent Reinforcement Learning

Jiahui Li [1] **Kun Kuang** [1] **Baoxiang Wang** [2 3] **Furui Liu** [4] **Long Chen** [1] **Changjie Fan** [5] **Fei Wu** [1 6 7] **Jun Xiao** [1]

Abstract

Value decomposition (VD) methods have been widely used in cooperative multi-agent reinforcement learning (MARL), where credit assignment plays an important role in guiding the agents' decentralized execution. In this paper, we investigate VD from a novel perspective of causal inference. We first show that the environment in existing VD methods is an unobserved confounder as the common cause factor of the global state and the joint value function, which leads to the confounding bias on learning credit assignment. We then present our approach, deconfounded value decomposition (DVD), which cuts off the backdoor confounding path from the global state to the joint value function. The cut is implemented by introducing the *trajectory graph*, which depends only on the local trajectories, as a proxy confounder. DVD is general enough to be applied to various VD methods, and extensive experiments show that DVD can consistently achieve significant performance gains over different state-of-the-art VD methods on StarCraft II and MACO benchmarks.



(a) Workflow of VD methods.

(b) Causal graph of VD methods.

(c) Causal graph of the proposed deconfounded VD method.

Figure 1. (a) The training procedure of VD methods, where E denotes the environment, s denotes the global state, $\tau = \{\tau_1, \dots, \tau_N\}$ denotes the local trajectories of all agents, $K = \{k_1, \dots, k_N\}$ denotes the credits of all agents, and Q_{tot} denotes the joint value function. The shaded area represents the central mixer, and the red line represents the credit assignment procedure. (b) The causal graph of VD method, where E is an unobserved confounder for learning credit assignment from s to Q_{tot}. (c) The proposed causal graph, where we introduce a new path from τ to K with the trajectory graph G as a proxy confounder for deconfounded training.

1. Introduction

Cooperative multi-agent reinforcement learning (MARL) has grown its popularity in many real-world applications, *e.g.*, traffic control (Schulman et al., 2016), robotics (Ramchurn et al., 2010; Lillicrap et al., 2016), scene understanding (Chen et al., 2019), and network routing (Ye et al., 2015). Limited by the partial observability and communication constraints, agents may have to make their own decisions based on local action-observation histories, which requires the learning of decentralized policies. To address this challenge, the paradigm of centralized training with decentralized execution (CTDE) (Oliehoek et al., 2008; Kraemer & Banerjee, 2016) has attracted unprecedented attention in which value decomposition methods (Rashid et al., 2018; Wang et al., 2021a;b) have shown strength on challenging tasks.

One of the main challenges in these methods is credit assignment which aims to deduce the contributions of individual agents from the overall success. It is an indispensable part for guiding the learning of decentralized policies and is usually designed as a module embedded in the central mixer. In

[1]DCD Lab, College of Computer Science, Zhejiang University. This work started when Long Chen at ZJU. [2]The Chinese University of Hong Kong, Shenzhen [3]Shenzhen Institute of Artificial Intelligence and Robotics for Society [4]Huawei Noah's Ark Lab [5]Fuxi AI Lab, NetEase Games [6]Shanghai Institute for Advanced Study of Zhejiang University [7]Shanghai AI Laboratory. Correspondence to: Kun Kuang <kunkuang@zju.edu.cn>.

Proceedings of the 39^{th} *International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

this paper, the process of credit assignment is investigated from a novel perspective of causal inference.

To illustrate the causal perspective, we provide a general workflow of value decomposition (VD) methods (Rashid et al., 2018; Wang et al., 2021a;a;b) in Figure 1(a). At each time step, the local action-observation trajectories $\tau = \{\tau_1, \dots, \tau_N\}$ as well as the global state s, are generated depending on the environment E. Then each local agent executes action u_i via the decentralized local action-value function $Q_i(\tau_i, u_i)$ which is then passed to the central mixer.[1] In the central mixer, credit assignment (red lines in Figure 1(a)) is performed to estimate the contribution of each agent $K = \{k_1, \dots, k_N\}$. Finally, the credits, as well as the local value functions $\{Q_1, \dots, Q_N\}$, consist of the joint value function Q_{tot}. In some methods, *e.g.*, QMIX (Rashid et al., 2018), $\{Q_1, \dots, Q_N\}$ will be further factorized into $\{\hat{Q}_1, \dots, \hat{Q}_M\}$, and correspondingly $K = \{k_1, \dots, k_M\}$ will be computed to estimate their contributions to Q_{tot}, as depicted in the dotted part in Figure 1(a). For a better analysis, we summarize this workflow as the causal graph in Figure 1(b).

Due to the fact that each agent observes the global state partially, the credits K are crucial for the decentralized value functions $Q_i(\tau_i, u_i)$ to accurately approximate $Q_i(s, u_i)$. However, the environment E is an unobserved confounder as the common cause factor of the global state and the joint value function. In fact, there is a backdoor path $s \leftarrow E \rightarrow \tau \rightarrow Q_{tot}$, which is harmful to traditional VD methods. Specifically, the backdoor path would bring a spurious correlation between the global state s and the joint value function Q_{tot}, leading to the confounding bias on learning credit assignment in which calculating $P(Q_{tot}|s)$ is involved. It thereby restricts the performance of the model. One possible approach to address the confounding bias is to calculate the do intervention $P(Q_{tot}|do(s))$ by cutting off the backdoor path (Pearl, 1995; Glymour et al., 2016). As the environment E is unobserved, it will be necessary to leverage the trajectories τ to achieve the cut. One will then obtain deconfounded credit assignment by calculating $P(Q_{tot}|do(s)) = \sum_{\tau} P(Q_{tot}|s, \tau)P(\tau)$. It is, however, intractable to estimate the right-hand side by sampling $\tau \sim P(\tau)$, as the environment is complicated and uncontrollable in general. As such, one will have to seek a different approach in MARL.

To achieve deconfounded training, motivated by the principle of backdoor adjustment in causal inference, we propose a new causal graph as shown in Figure 1(c). The new graph sets up a new path $\tau \rightarrow G \rightarrow K$ to estimate the credits K with a new variable G that depends only on τ. The new path decomposes the confounding bias on learning credits assignment into two parts: one is from the backdoor path $s \leftarrow E \rightarrow \tau \rightarrow G \rightarrow K$, and two is from $K \leftarrow G \leftarrow \tau \rightarrow Q_{tot}$. In this way, the variable G can serve as a proxy confounder to cut off these two backdoor paths and realize $P(K|do(s))$ and $P(Q_{tot}|do(K))$. Based on the proposed causal graph, we propose deconfounded value decomposition (DVD). DVD implements G with the trajectory graph, which connects the hidden states of the trajectory of each agent. This realizes deconfounded credit assignment by achieving $P(Q_{tot}|do(s))$ with calculating both $P(K|do(s))$ and $P(Q_{tot}|do(K))$.

DVD is general enough to be applied to various VD methods, such as QMIX (Rashid et al., 2018), QPLEX (Wang et al., 2021a), and RODE (Son et al., 2019). The efficacy and compatibility of DVD has been verified by extensive experiments on StarCraft II (Samvelyan et al., 2019) and MACO (Wang et al., 2020) benchmarks. With DVD, each of QMIX, QPLEX, and RODE enjoys a significant improvement on each benchmark task. The improvement is especially large when the credit assignment plays an important role in the task.

Our main contributions can be summarized as follows: (i) We are the first to develop a causal perspective of value decomposition in MARL and the first to point out the confounding bias in learning credit assignment. (ii) We propose deconfounded value decomposition (DVD), a new framework motivated by the principle of backdoor adjustment to remove such bias. (iii) DVD is compatible with existing VD methods and introduce large improvements to them.

2. Related Work

The joint action space in multi-agent reinforcement learning grows exponentially with the number of the participated agents which promotes the development of the paradigm of centralized training with decentralized execution (CTDE). Under CTDE, value decomposition methods (Rashid et al., 2018; Son et al., 2019; Yang et al., 2020a; Wang et al., 2021a;b; Zhang et al., 2021) show their strength in expressing the joint value function conditioned on global trajectory via individual value functions conditioned on the local observation-action history. One of the major challenges in CTDE is credit assignment, which aims to deduce the contributions of each individual agent to the overall success.

Explicit credit assignment methods (Foerster et al., 2018; Yang et al., 2020a; Li et al., 2021) train the central critic and the local agents separately and the credits are computed via a designed algorithm or module. COMA (Foerster et al., 2018) leverages the difference between the joint value function and a counterfactual baseline to represent the local contributions. QPD (Yang et al., 2020a) computes the credits according to the integrated gradients of the inputs in the mixer. Li (Li et al., 2021) proposes a more algorithmic

[1] For simplicity, we shorthand $Q_i(\tau_i, u_i)$ as Q_i in Figure 1(a).

method that utilizes Shapley value to infer the credits fairly.

We focus on implicit methods, where the credit assignment procedure is performed in a module inside the central critic, usually a network. These methods are designed under the assumption of Individual Global-Max (IGM), which guarantees the consistency between individual optimal actions and optimal joint action. VDN (Sunehag et al., 2018) and QMIX (Rashid et al., 2018) provide sufficient conditions for IGM by additivity and monotonicity. QPLEX (Wang et al., 2021a) utilizes duplex dueling architecture to alleviate the sub-optimal phenomena induced by the restriction of monotonicity. FOP (Zhang et al., 2021) introduces a more general condition Individual-Global-Optimal and proposes a novel method to factorize the joint policy induced by maximum-entropy MARL into individual policies.

However, these methods neglect a fact that the direct causal effect of the assignment module is confounded by the environment which further limits the performance of the model. We propose a novel method to address this issue via the backdoor adjustment in this paper.

3. Preliminaries

3.1. Dec-POMDPs

We focus on fully cooperative multi-agent tasks with the settings of decentralised partially observable Markov decision process (Dec-POMDP) (Oliehoek & Amato, 2016; Bernstein et al., 2002; Busoniu et al., 2008; Gupta et al., 2017; Palmer et al., 2018), which can be modeled as a tuple:

$$G = < N, S, U, P, r, Z, O, \gamma >,$$

where N represents the set of agents with $|N| = N$, and $s \in S$ represents the global state of the environment. At each time step, each agent $i \in N$ chooses an action $u_i \in U$ to formulate a joint action $u \in U^N$. This joint action results in a state transition on the environment according to the transition function $P(s'|s, u) : S \times U^N \rightarrow S$. Then, all agents receive a shared reward function according to the reward function $r(s, u) : S \times U^N \rightarrow \mathbb{R}$. Moreover, each agent only has access to a partial observation $z \in Z$ which are generated by an observation function $O(s, i)$: $S \times N \rightarrow Z$. Each agent learns its own policy $\pi^i(u_i|\tau_i) : T \times U \rightarrow [0, 1]$ conditions on the its local trajectory $\tau_i \in T$. The objective of all agents is to maximize the discounted cumulative return $\sum_{i=0}^{\infty} \gamma^i r_i$, where γ is a discount factor.

The environment E acts as the confounder in value decomposition methods which is a crucial concept in this paper, and we define it as the agents' world in which they live and interact, and this includes the neural networks.

3.2. Credit Assignment in VD Methods

Value decomposition methods are the most popular branches under the framework of CTDE. In these methods, credit assignment aims to infer the contributions of predecessor value functions to the joint value function Q_{tot}, and such a procedure is usually performed in a human-designed module. We represent it as a more general formulation:

$$Q_{tot} = \sum_{j=1}^{M} k_j \hat{Q}_j, \tag{1}$$

where $\hat{Q}_j$ represents the predecessor value functions and M denotes its number. k_j represents the credits that reflect the contributions of each value function to the total benefits. M and $\hat{Q}_j$ are various in different methods. For example, in QMIX (Rashid et al., 2018) M depends on the hyperparameter and $\hat{Q}_j$ represents the output of a model layer, while in QPLEX (Wang et al., 2021a) $M = N$ and $\hat{Q}_j$ denotes the decentralized local value functions. We will justify this equation in *Appendix* A.

4. Method

4.1. Backdoor Adjustment for Deconfounded Training

As the causal graph in Figure 1(b) shows, the unobserved environment E influences the generation of the global state s and the local trajectories τ, and then τ is the direct cause of joint value function Q_{tot}. Hence, it creates a backdoor path from s to Q_{tot}, i.e., $s \leftarrow E \rightarrow \tau \rightarrow Q_{tot}$, which would bring spurious correlation between s and Q_{tot}, resulting in confounding bias on learning credit assignment procedure $s \rightarrow K \rightarrow Q_{tot}$. From the causal literature (Pearl, 1995; Zhang et al., 2020; Yang et al., 2021), the backdoor adjustment is one possible way to address the confounding bias, and applying causal intervention based on the specified confounders will help cut off the backdoor path. From Figure 1(b), we know the environment E is an appropriate confounder that can fully cut off all backdoor paths from s to Q_{tot}, but E is hardly observed in real applications. Fortunately, the local trajectories τ can also serve as the role of confounder to realize deconfounded training via backdoor adjustment with the following equation:

$$P(Q_{tot}|do(s)) = \sum_{\tau} P(Q_{tot}|s, \tau)P(\tau), \tag{2}$$

where $\tau \in T^N$ denotes the the local trajectories of all agents, and $P(Q_{tot}|s)$ denotes the prediction of the path $s \rightarrow K \rightarrow Q_{tot}$.

Such a method relies on the help of another variable τ. However, as the environment is complicated and can not be modeled, we cannot get or even sample τ. Thus, it is prohibitively to achieve the above backdoor adjustment.

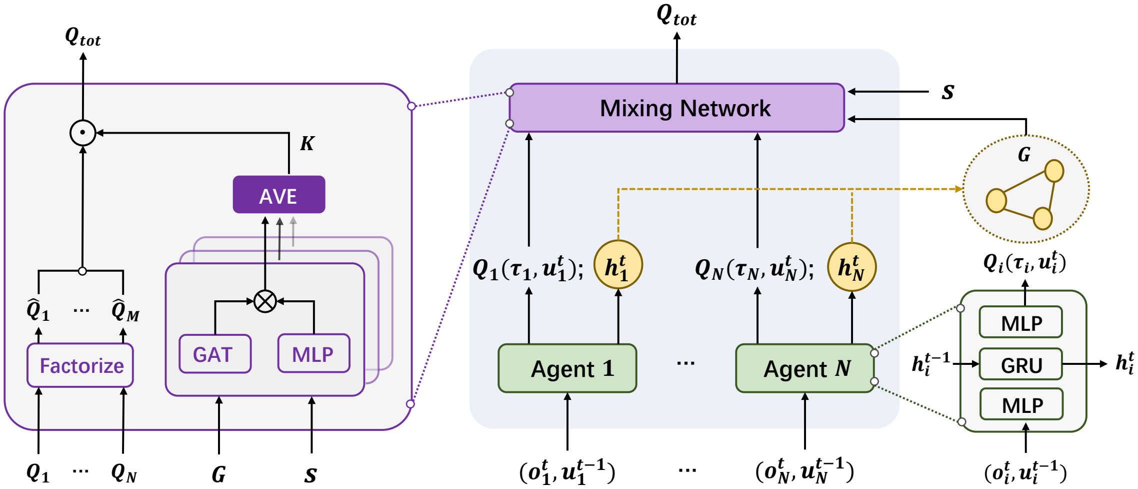


Figure 2. The framework of our method. First, each local agent models a value function conditions on its local observation-action history. Then, we construct a trajectory graph via hidden states in their RNNs. In the mixing network, local value functions $\{Q_1, \dots, Q_N\}$ will be factorized into $\{\hat{Q}_1, \dots, \hat{Q}_M\}$, and the graph as well as the global state are used to estimate the credits. Finally, the joint value function is computed via credits K and factorized value functions $\{\hat{Q}_1, \dots, \hat{Q}_M\}$. The whole framework is trained via TD-loss.

To this end, we propose a novel causal structure which is shown in Figure 1(c) by setting up a new path $\tau \to G \to K$ to connect τ and K, where G is the intermediate node and only depends on τ. We implement G via *trajectory graph* which will be introduced in details in Section 4.2. By connecting τ and K, one can estimate the credits K with both s and τ, where τ contains more history information, thereby bringing more precise prediction on credits K. Moreover, constructing the *trajectory graph* G based on τ allows message exchanging among agents, which further promotes the prediction of K. From the causal view, the newly added path $\tau \to G \to K$ can help decompose the confounding bias on learning credits assignment into two parts: one is from the backdoor path $s \leftarrow E \to \tau \to G \to K$, and the other is from $K \leftarrow G \leftarrow \tau \to Q_{tot}$. Fortunately, the *trajectory graph* G can serve as a proxy confounder to cut off these two back paths and achieve deconfounded credit assignment for MARL.

From the Figure 1(c), we know the credits K is the only mediator of causal path $s \to K \to Q_{tot}$ on learning credit assignment. Although directly calculating $P(Q_{tot}|do(s))$ like Eq. (2) is impractical, we propose to realize deconfounded credit assignment by achieving $P(Q_{tot}|do(s))$ with calculating both $P(K|do(s))$ and $P(Q_{tot}|do(K))$. With the newly introduced trajectory graph G as a proxy confounder, fortunately, one can cut off the backdoor paths $s \leftarrow E \to \tau \to G \to K$ and $K \leftarrow G \leftarrow \tau \to Q_{tot}$ in Figure 1(c) for calculating both $P(K|do(s))$ and $P(Q_{tot}|do(K))$ via backdoor adjustment with the following equations:

$$P(K|do(s)) = \sum_G P(K|s, G)P(G), \tag{3}$$

$$P(Q_{tot}|do(K)) = \sum_G P(Q_{tot}|K, G)P(G). \tag{4}$$

Sometimes, we may encounter the situations where there are infinite number of G. Hence, it is necessary to replace Eq. (3) and Eq. (4) with its approximation.

We approximate the backdoor adjustment in Eq. (3) via Monte Carlo sampling on the trajectory graph G as follows:

$$P(K|do(s)) \approx \frac{1}{D_1}\sum_{d=1}^{D_1} P(K|s, G^d), \tag{5}$$

where D_1 represents the number of sampling times, and G^d refers to the sampled graph at times d.

Similarity, we approximate the backdoor adjustment in Eq. (4) with Monte Carlo sampling as follows:

$$P(Q_{tot}|do(K)) \approx \frac{1}{D_2}\sum_{d=1}^{D_2} P(Q_{tot}|K, G^d), \tag{6}$$

where D_2 represents the number of sampling times.

From Figure 1(c), we know the credit K depends on both the trajectory graph G and the global state s. Given the global state s, the credit K can be calculated with the sampled G^d, denoted as K^d. Therefore, we can further approximate $P(Q_{tot}|K, G^d)$ by calculating $P(Q_{tot}|K^d, G^d)$, and estimating $P(Q_{tot}|do(K))$ as:

$$P(Q_{tot}|do(K)) \approx \frac{1}{D_2}\sum_{d=1}^{D_2} P(Q_{tot}|K^d, G^d). \tag{7}$$

We set $D_1 = D_2 = D$ in this paper, and the implementation details will be discussed in the next subsection.

4.2. Implementation

Based on the previous analysis, we propose a novel method to implement the deconfounded training. The whole training procedure is shown in Figure 2. First, each agent models a local value function. Then, we extract all of the agents' representations to construct a *trajectory graph* at each time step, which will be further passed to the central mixer. Afterward, both the graph and the global state are utilized to estimate the credits. Finally, the joint value functions are predicted depending on the credits as well as the local value functions.

Construction of Trajectory Graph G. We represent the local value functions at time step t as $Q_i^t = f_i(h_i(\tau_i^t))$, where $h_i(\cdot)$ denotes the hidden states of the Recurrent Neural Network, and $f_i(\cdot)$ denotes the inference layer for local value functions. For simplicity, we shorthand $h_i(\tau_i^t)$ as h_i. Then, we construct the *trajectory graph* $G = < V, E >$, where $V = \{h_i, ..., h_N\}$ are the nodes and $E = \{< h_i, h_j > | i \neq j\}$ denotes the edges of the graph. Namely, we construct G by setting each hidden states h_i as a node and connecting any two nodes as an edge.

Deconfounded Credit Assignments. To accomplish the backdoor adjustment in Eq. (5) & (7), we need to sample several G at each time step. For alleviating the computational burden and finishing backdoor adjustment in one forward pass, we apply a multi-head strategy where the trajectory graph G is utilized to generate multiple representations $\{G^1, ..., G^D\}$. Concretely, for each representation of the trajectory graph G^d, we model a Graph Attention Layer (Veličković et al., 2018) for exchanging messages among nodes. Each node is updated as:

$$h_i' = \sigma\Big(\sum_{j \in N_i} \alpha_{ij} W h_j\Big), \tag{8}$$

where α_{ij} denotes the attention weights of different nodes, and W denotes the common weight matrix. The attention weights is computed as:

$$\alpha_{ij} = \frac{exp(e_{ij}))}{\sum_{m \in N_i} exp(e_{im}))}, \tag{9}$$

$$e_{ij} = LeakReLU(a[Wh_i || Wh_j]), \tag{10}$$

where a is a single feed forward network.

Hence, we get the representation of the trajectory graph $G^d = \{h_1', ..., h_N'\}$ which can be represented by a matrix. Based on the Eq. (5), then, we implement the corresponding $P(K|s, G^d)$ (denoted as K^d) with:

$$K^d := P(K|s, G^d) = |f_s(s) G^d|, \tag{11}$$

where f_s is the function that map s to a representation matrix, and $| \; |$ represents the absolute value symbol which is used to fulfill the monotonicity constraint (Rashid et al., 2018).

Then, the credits $K = \{k_1, \dots, k_n\}$ is computed as:

$$K = \frac{1}{D} \sum_{d=1}^{D} K^d, \tag{12}$$

which completes the backdoor adjustment $P(K|do(s))$.

Similarity, we implement the backdoor adjustment in Eq. (7) with the sampled $K^d = \{k_1^d, \dots, k_n^d\}$ as:

$$Q_{tot} = \frac{1}{D} \sum_{d=1}^{D} \sum_{j=1}^{M} k_j^d \hat{Q}_j = \sum_{j=1}^{M} k_j \hat{Q}_j, \tag{13}$$

which completes the backdoor adjustment $P(Q_{tot}|do(K))$.

So far, we achieve the deconfounded training on $P(Q_{tot}|do(s))$ by calculating $P(K|do(s))$ and $P(Q_{tot}|do(K))$ with Eq. (12) & (13), respectively.

The whole framework is trained via TD-loss: $L(\theta)$:

$$\begin{aligned} L(\theta) &= (Q_{tot} - y)^2, \\ y &= r + \gamma Q_{tot}, \end{aligned} \tag{14}$$

where θ represents the parameters of the whole network, and Q_{tot} represents the output of the target network.

We demonstrate the details of our algorithm in *Algorithm* 1.

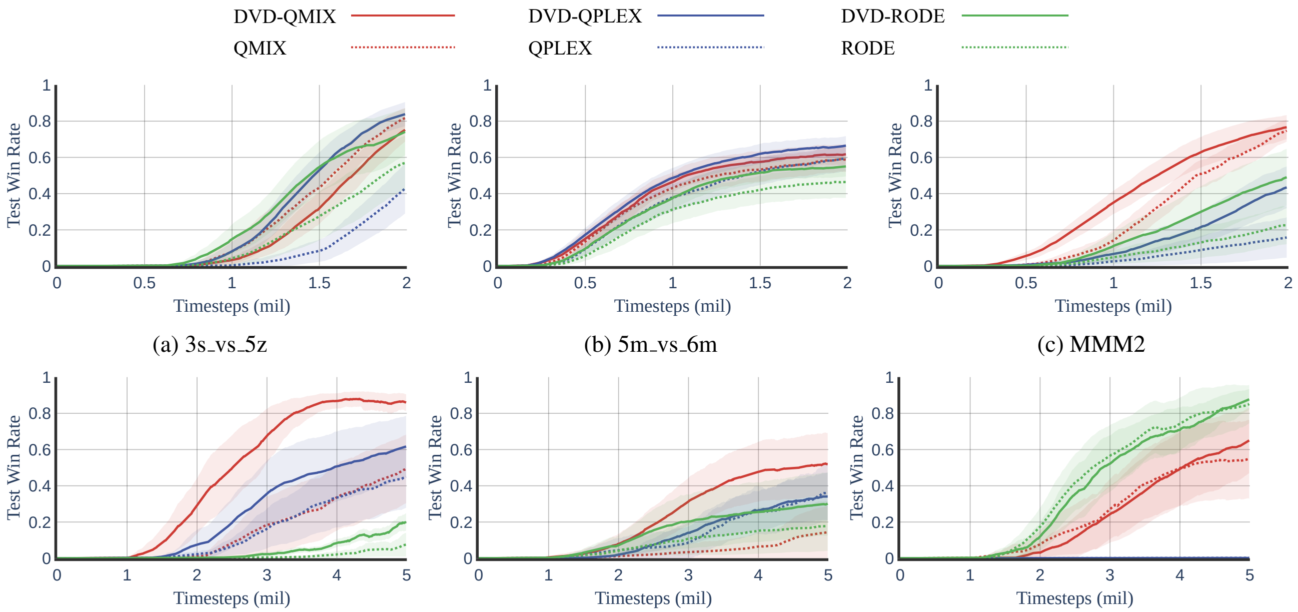
5. Experiment

To evaluate the effectiveness of the deconfounded training, we apply our approach to three popular value decomposition baselines, including QMIX (Rashid et al., 2018), QPLEX (Wang et al., 2021a) and RODE (Wang et al., 2021b). We show that each baseline enjoys a significant improvement.

5.1. Experimental Setup

We carry out the experiments with different scenarios on two benchmarks, StarCraft II micro management challenge (SMAC) (Samvelyan et al., 2019) and multi-agent coordination challenge (MACO) (Wang et al., 2022).

StarCraft II Micro Management Benchmark. StarCraft II is a real-time strategy game, and in the micro management



(a) 3s_vs_5z

(b) 5m_vs_6m

(c) MMM2

(d) 3s5z_vs_3s6z

(e) 6h_vs_8z

(f) corridor

Figure 3. Performance comparison with baselines on the StarCraft II micro management benchmark.

challenge the agents need to cooperate with each other to battle with the opponent's armies which are controlled by the hand-coded built-in AI. This benchmark consists of various maps which have been classified as easy, hard, and super hard. We set the difficulty of the game AI to the "very difficult" level. The experiments are performed on three hard maps (3s_vs_5z, 5m_vs_6m, MMM2) and three super hard maps (3s5z_vs_3s6z, corridor, 6h_vs_8z).

Multi-Agent Coordination Challenge Benchmark. MACO integrates the classic coordination tasks from different multi-agent learning literature and improves their difficulty. It contains 6 representative problems: *Aloha*, *Pursuit*, *Hallway*, *Sensor*, *Gather* and *Disperse* in which different degrees of coordination among agents is needed. Since the action space is not very large in the MACO benchmark, RODE is not suitable for these scenes. Thus, we only apply our method on QMIX and QPLEX. For the details about the two benchmarks, please refer to *Appendix* B.1.

5.2. Performance on StarCraft II

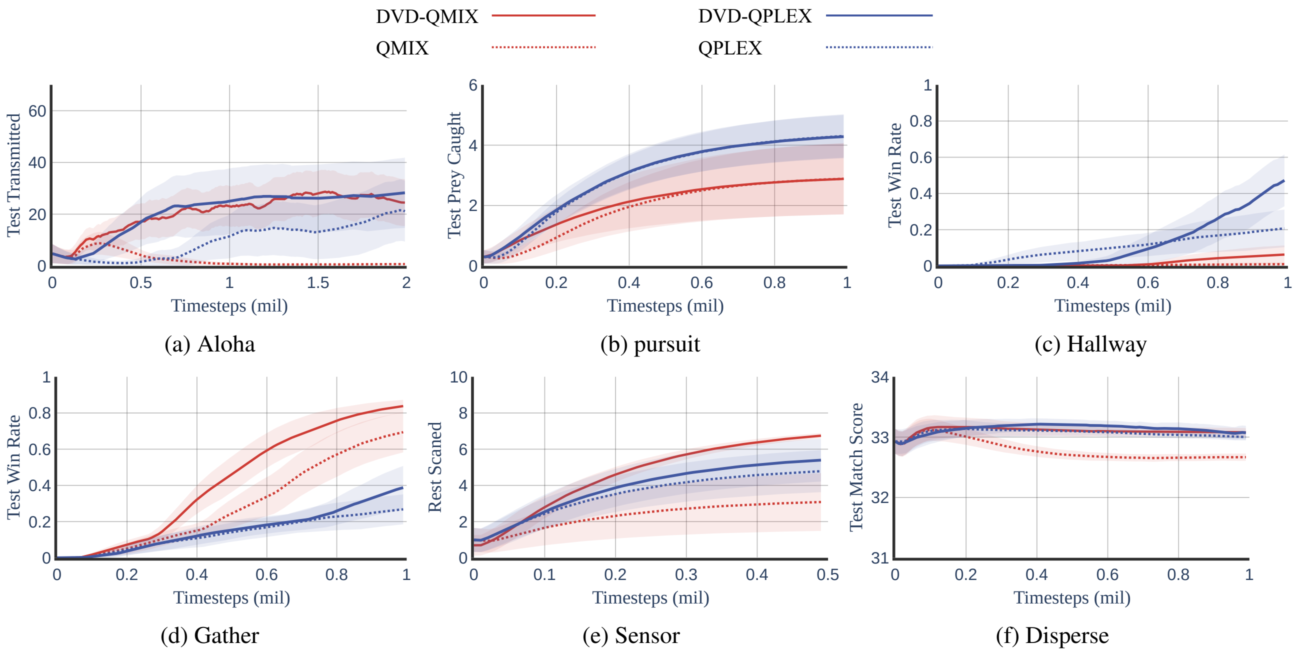
We carry out the experiments on StarCraft II with 10 random seeds, and the average results are shown in Figure 3.

In map 3s_vs_5z, 3 *Stalkers* need to fight with 5 *Zealots*. Since *Zealots* counter *Stalkers* in attack and armor types, the only way to win this game is to kite the enemy. That means, the agents need to balance the actions of "move" and "fire". The deconfounded training make a huge improvement on QPLEX and RODE and a small drop on QMIX. We suspect that QMIX has already gained a good performance and the confounding bias is relatively small in this task. A more complicated model will then hinder its original performance.

In map 5m_vs_6m, where the ally has 5 *Marines* and the enemy has 6 *Marines*, each agent needs to focus on beating enemies and avoid taking redundant actions. Meanwhile, a little kiting strategy is also needed. We can see that all of the baselines get an improvement by applying the deconfounded training.

In map MMM2, 1 *Medivac*, 2 *Marauders*, and 7 *Marines* are fighting with a stronger enemy team which consists of 1 *Medivac*, 3 *Marauders*, and 8 *Marines*. Among these units, *Medivac* is the most special one because it can heal the injured allies. In this scenario, credit assignment plays a more important role, and our method improves the three baselines in varying degrees.

Our method outperforms the baselines significantly in map 3s5z_vs_3s6z, where the ally has 3 *Stalkers* and 5 *Zealots* while the enemy has 3 *Stalkers* and 6 *Zealots*. Especially, we improve the mean win rate of QMIX by nearly 40 percent and we get close to perfect (100 percent mean win rates). We watched several game replays and found that, one of the ally units learned to sacrifice itself to kite 4 to 5 enemy units to the corner of the map. During this time, the other allied units kill the remained enemy with little damage taken. Then the enemy who had been led away are wiped out easily. Obviously, for the special agent, running away or dying is more valuable than attacking. This pattern can be learned



(a) Aloha (b) pursuit (c) Hallway

(d) Gather (e) Sensor (f) Disperse

Figure 4. Performance comparison with baselines on the multi-agent coordination challenge benchmark.

only when credit assignment guides the decentralized agents correctly, which affirms the importance of deconfounded training, and explains the reason for the efficiency of our method.

Map 6h_vs_8z is more challenging, where the ally has 6 *Hydraliks* and the enemy has 8 *Zealots*. The agents need to kite the enemy in turn as well as focus on firing to win the game. We improve QMIX almost 5 times performance from 10 mean win rates to 50 mean win rates.

Corridor is the most special map where the ally has 6 *Zealots* and the enemy has 24 *Zerglings*. In this scenario, QPLEX behaves poorly and can not learn any useful pattern. There is almost no improvement of our method which can be treated as a failure case. The possible reason for the failure of our method is that the credit assignment procedure is not very critical in this map. RODE and DVD-RODE demonstrate a stronger performance in this scenario which means the transformation of the action space plays a more important role than the other tricks.

5.3. Performance on MACO

The results are shown in Figure 4, where the lines are the average performance with 10 random seeds and the shadow areas represent the standard deviation.

In *Aloha*, the learning curve of QMIX corrupts after 0.5 million steps, and our method improves nearly 30 transmissions, which means there is a serious spurious relationship between credits and joint value function in this scenario. As for QPLEX, we also improve it by nearly 10 transmissions.

Since there is no communication among agents in *Pursuit* in our settings, the credit assignment procedure does not work. Hence, there is no improvement of backdoor adjustment. Even though, our method convergences a little faster which benefits from the *trajectory graph* as it performs message exchanging in the mixer.

The main challenge in *Hallway* is that groups of agents need to reach a target state at the same time but they don't know the information of the others. Wang *et al*. (Wang et al., 2022) has pointed out that fully-decomposed value decomposition methods cannot solve this problem if the initial positions of agents are stochastic. But our method still shows its strength as it improves the performance of QPLEX by nearly 40 percent mean win rates. QMIX behaves poorly in this scenario not only because of the confounding effect but its own capacity, thus we only get a slight improvement.

Credit assignment plays important role in *Sensor*, since local agents need to know whether their scans are the meaningful scans that bring the rewards. Our method doubled the performance of QMIX. We watch the replay and find that most scanned targets sit in the middle of four agents. This is a conservative strategy, because each target can be scanned by four agents simultaneously.

In *Gather*, each agent has a high risk to fall into suboptimal, since a wrong action will also lead to a reward. Thanks to the deconfounded training, the role of credit assignment is fully utilized, and we achieve nearly 90 percent mean win

Algorithm 1 Deconfounded Value Decomposition

```
Initialize: Networks of local agents and central mixer  , target networks ~, max episode length $T$
  for each training episode do
    while global state $s \neq$ terminal and time step $t < T$ do
      $t = t + 1$
      for each agent $i$ do
        Compute the local value function $Q_i$ and get the hidden state $h_t^i$
        Select action $u_t^i$ via value function and exploration strategy
      end for
      Execute the joint action $(u_t^1; u_t^2; :::; u_t^n)$
      Get reward $r_{t+1}$ and next state $s_{t+1}$
    end while
    Add episode to replay buffer
    Collate episodes in buffer into a single batch
    for $t = 1$ to $T$ do
      Construct trajectory graph $G$ via the hidden states $fh_t^1; : : : ; h_t^N g$
      Compute the credits via Eq. (11) and Eq. (12)
      Compute the joint value function via Eq. (13)
      Compute the targets $y$ using central target network
      Update   by minimizing the loss $L(\ )$ defined in Eq. (14)
      Update ~ =   periodically
    end for
  end for
```

rates which improves QIMX a huge gap.

There is only a little improvement of our method to the baseline in *Disperse*. That is because *Disperse* is too hard for no communication settings, and all the methods can not learn an efficient pattern.

5.4. Ablation Study

Our ablation study aims to answer the following questions: (1) Whether the created new path is rational? (2) How does the model performance benefits from the deconfounded training? (3) How do the sampling times influence the efficiency of backdoor adjustment? We perform ablations on map MMM2 in the StarCraft II benchmark since it is a hard and representative scenario. We choose QPLEX as the baseline method. The results are shown in Figure 5, where "h1", "h4", "h8" represent the hyper-parameter $D = 1$, $D = 4$, and $D = 8$ separately.

When we set $D = 1$ (the green line in the figure), it improves the baseline a little. Since there is only one-time sampling, backdoor adjustment can not be performed and the credit assignment procedure still suffers from the confounding effect caused by the environment. Such an improvement comes from the message exchanging among agents brought by trajectory graph which makes the prediction of credits K more precise. The result also demonstrates the rationality of the path $\ ! \ G \ ! \ K$.

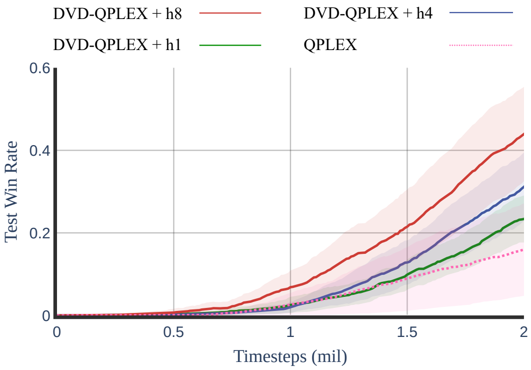


Figure 5. Ablation study with different sampling times for backdoor adjustment. The experiment is performed on map MMM2 in the StarCraft II benchmark. QPLEX is chosen as the baseline.

When we set $D = 4$, there is a further improvement on the performance (the blue line in the figure). The explanation of the gap between the green line and the blue line is that how does the deconfounded training make sense when the path is already set up. In order to find a moderate hyper-parameter we increase D gradually and when $D = 8$ the best performance is achieved.

To be mentioned that, the most import role of G is to serve as the proxy confounder, and it can be implemented by any rational form. We implement G with graph attention network (Veličković et al., 2018) for its efficiency. To justify that the performance comes from the deconfounded training not the graph structure, We realize G with other forms as well as compare DVD with the other graph-based methods in MARL. The results are left in *Appendix* C.

6. Conclusion

In this paper, we investigate the value decomposition methods in multi-agent reinforcement learning with a causal perspective. We find that most of the mainstream VD methods suffer from confounding bias on learning credit assignment. The bias is brought by the unobserved environment in MARL and makes the model cannot exert its due ability. To address the bias and achieve the deconfounded training, we propose a novel method named deconfounded value decomposition (DVD). Concretely, we construct a trajectory graph via local trajectories. By treating the trajectory graph as a proxy confounder, we perform do intervention that cuts off all the backdoor paths along with the environment. Our DVD method can be applied to various VD methods, and the experiments show the superiority of DVD.

7. Acknowledgements

This work was supported by the National Key Research & Development Project of China (2021ZD0110700), National Key Research and Development Program of China (No. 2018AAA0101900), Key R & D Projects of the Ministry of Science and Technology (2020YFC0832500), Young Elite Scientists Sponsorship Program by CAST (2021QNRC001), National Natural Science Foundation of China (U19B2043, 61976185, No. 62006207, No. 62037001), Project by Shanghai AI Laboratory (P22KS00111), Zhejiang Natural Science Foundation (LR19F020002), Zhejiang Innovation Foundation(2019R52002), and the Fundamental Research Funds for the Central Universities(226-2022-00142). Baoxiang Wang is partially supported by The National Natural Science Foundation of China (62106213 and 72150002) and AIRS (AC01202101108 and AC01202101031).

References

Benda, M. On optimal cooperation of knowledge sources: an empirical investigation. *Technical Report, Boeing Advanced Technology Center*, 1986.

Bernstein, D. S., Givan, R., Immerman, N., and Zilberstein, S. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

Böhmer, W., Kurin, V., and Whiteson, S. Deep coordination graphs. In *International Conference on Machine Learning*, pp. 980–991. PMLR, 2020.

Busoniu, L., Babuska, R., and De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

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.

Cheng, S. *Coordinating decentralized learning and conflict resolution across agent boundaries*. PhD thesis, The University of North Carolina at Charlotte, 2012.

Crick, C. and Pfeffer, A. Loopy belief propagation as a basis for communication in sensor networks. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 159–166, 2002.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

Guestrin, C., Lagoudakis, M., and Parr, R. Coordinated reinforcement learning. In *ICML*, volume 2, pp. 227–234. Citeseer, 2002.

Gupta, J. K., Egorov, M., and Kochenderfer, M. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 66–83, 2017.

Hansen, E. A., Bernstein, D. S., and Zilberstein, S. Dynamic programming for partially observable stochastic games. In *AAAI*, volume 4, pp. 709–715, 2004.

Kraemer, L. and Banerjee, B. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

Lesser, V., Ortiz Jr, C. L., and Tambe, M. *Distributed sensor networks: A multiagent perspective*, volume 9. Springer Science & Business Media, 2003.

Li, J., Kuang, K., Wang, B., Liu, F., Chen, L., Wu, F., and Xiao, J. Shapley counterfactual credits for multi-agent reinforcement learning. *KDD*, 2021.

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. *ICLR*, 2016.

Murphy, K. P., Weiss, Y., and Jordan, M. I. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 467–475, 1999.

Naderializadeh, N., Hung, F. H., Soleyman, S., and Khosla, D. Graph convolutional value decomposition in multi-agent reinforcement learning. *arXiv preprint arXiv:2010.04740*, 2020.

Oliehoek, F. *Value-based planning for teams of agents in stochastic partially observable environments*. Amsterdam University Press, 2010.

Oliehoek, F. A. and Amato, C. *A concise introduction to decentralized POMDPs*. 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.

Palmer, G., Tuyls, K., Bloembergen, D., and Savani, R. Lenient multi-agent deep reinforcement learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 443–451, 2018.

Pearl, J. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan kaufmann, 1988.

Pearl, J. Causal diagrams for empirical research. Biometrika, 82(4):669–688, 1995.

Ramchurn, S. D., Farinelli, A., Macarthur, K. S., and Jennings, N. R. Decentralized coordination in robocup rescue. The Computer Journal, 53(9):1447–1461, 2010.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In International Conference on Machine Learning, pp. 4295–4304, 2018.

Samvelyan, M., Rashid, T., De Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. The starcraft multi-agent challenge. 2019.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. ICLR, 2016.

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 International Conference on Machine Learning, pp. 5887–5896, 2019.

Stone, P. and Veloso, M. Multiagent systems: A survey from a machine learning perspective. Autonomous Robots, 8 (3):345–383, 2000.

Stranders, R., Farinelli, A., Rogers, A., and Jennings, N. R. Decentralised coordination of mobile sensors using the max-sum algorithm. In Twenty-First International Joint Conference on Artificial Intelligence, 2009.

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 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2085–2087, 2018.

Velicković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In International Conference on Learning Representations, 2018.

Wang, J., Ren, Z., Liu, T., Yu, Y., and Zhang, C. Qplex: Duplex dueling multi-agent q-learning. International Conference on Learning Representations, 2021a.

Wang, T., Wang, J., Zheng, C., and Zhang, C. Learning nearly decomposable value functions via communication minimization. In International Conference on Learning Representations, 2020.

Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., and Zhang, C. Rode: Learning roles to decompose multi-agent tasks. 2021b.

Wang, T., Zeng, L., Dong, W., Yang, Q., Yu, Y., and Zhang, C. Context-aware sparse deep coordination graphs. 2022.

Wei, E. and Luke, S. Lenient learning in independent-learner stochastic cooperative games. The Journal of Machine Learning Research, 17(1):2914–2955, 2016.

Yang, X., Zhang, H., and Cai, J. Deconfounded image captioning: A causal retrospect. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

Yang, Y., Hao, J., Chen, G., Tang, H., Chen, Y., Hu, Y., Fan, C., and Wei, Z. Q-value path decomposition for deep multiagent reinforcement learning. International Conference on Machine Learning, pp. 10706–10715, 2020a.

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, 2020b.

Ye, D., Zhang, M., and Yang, Y. A multi-agent framework for packet routing in wireless sensor networks. sensors, 15(5):10026–10047, 2015.

Yedidia, J. S., Freeman, W. T., Weiss, Y., et al. Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium, 8(236-239): 0018–9448, 2003.

Zhang, C. and Lesser, V. Coordinated multi-agent reinforcement learning in networked distributed pomdps. In Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

Zhang, S., Jiang, T., Wang, T., Kuang, K., Zhao, Z., Zhu, J., Yu, J., Yang, H., and Wu, F. Devlbert: Learning deconfounded visio-linguistic representations. In MM '20: The 28th ACM International Conference on Multimedia, pp. 4373–4382. ACM, 2020.

Zhang, T., Li, Y., Wang, C., Xie, G., and Lu, Z. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. International Conference on Machine Learning, pp. 12491–12500. PMLR, 2021.

A. Credit Assignment in Value Decomposition Methods

We have defined a general formulation of credit assignment procedure of value decomposition methods in Section 3:

$$Q_{tot} = \sum_{j=1}^{M} k_j \hat{Q}_j ; \tag{15}$$

where $\hat{Q}_j$ represents the predecessor value functions and M denotes its number. k_j represents the credits that reflect the contributions of each value function to the total benefits. Here, we will justify that all of the three baselines we used can be summarized as this formulation. For convenience, all of the bias networks are omitted.

The first baseline is QMIX (Rashid et al., 2018), where the joint value function is composed in two steps. At the first step, a neural network $f_w(\cdot)$ is set up to compute the weights for local value functions $Q_{local} = [Q_1; \ldots; Q_N]$:

$$W = |f_w(s)|; \tag{16}$$

where $W \in \mathbb{R}^{N \times Emb}$, N is the number of agents, and Emb is the hyper-parameter. Then, the local value functions are factorized into $Q_{inter} = [\hat{Q}_1; \ldots; \hat{Q}_{Emb}]$ via:

$$Q_{inter} = W Q_{local} : \tag{17}$$

At the second step, another neural network $f_k(\cdot)$ is set up to compute the credits $K = [k_1; \ldots; k_{Emb}]$ for Q_{inter} :

$$K = f_k(s): \tag{18}$$

Finally, the joint value function is computed via:

$$Q_{tot} = Q_{inter} K^{\top} ; \tag{19}$$

which is equivalent to Eq. (15) when $M = Emb$.

Similarly, the second baseline QPLEX (Wang et al., 2021a) estimates the credits $K = [k_1; \ldots; k_N]$ via a network $f_k(\cdot)$:

$$K = f_k(s): \tag{20}$$

And the joint value function is represented as:

$$Q_{tot} = \sum_{i=1}^{N} k_i Q_i + k_i A_i ; \tag{21}$$

where Q_i represents the local value function, and A_i represents the advantage function. Thus, QPLEX is equivalent to Eq. (15) when $M = N$ and $\hat{Q}_i = Q_i + A_i$ for $i = 1$ to N.

The last baseline RODE (Wang et al., 2021a) improves QMIX by reducing the action space of each agent via role networks. Except for the role network, the remaining framework of RODE is the same as QMIX, thus fulfilling Eq. (15).

B. Experiment Settings

B.1. Benchmark

StarCraft Multi-Agent Challenge In the StarCraft II micromanagement challenge (SMAC) (Samvelyan et al., 2019), each allied army is controlled by an agent and acts based on its local observations. The opponent's armies are controlled by the hand-coded built-in StarCraft II AI. The goal is to kill all enemies for each battle scenario. The environment produces rewards based on the hit-point damage dealt and enemy units killed. Besides, another bonus is given when the battle wins. At each time step, each agent only gets the observations within its field of view which includes the other alive agents. Besides, all agents can only attack the enemies within their shooting range, which is set to 6. The global state s consists of the joint observations without the restriction of the sight range, which will be used to predict credits K in the central mixer. Table 1 briefly introduces the SMAC challenges used in our paper, in which 3s_vs_5z, 5m_vs_6m, and MMM2 are hard maps while 3s5z_vs_3s6z, 6h_vs_8z, and corridor are super hard maps.

.

Table 1. Experiment maps of StarCraft II micro management benchmark.

Map Name	Ally Units	Eenemy Units	Agent Type
3s_vs_5z	3 Stalkers	5 Zealots	Homogeneous
5m_vs_6m	5 Marines	6 Marines	Homogeneous
MMM2	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 3 Marauders, 8 Marines	Heterogeneous
3s5z_vs_3s6z	3 Stalkers, 5 Zealots	3 Stalkers, 6 Zealots	Heterogeneous
6h_vs_8z	6 Hydraliks	8 Zealots	Homogeneous
corridor	6 Zealots	24 Zerglings	Homogeneous

Multi-Agent Coordination Challenge Multi-Agent COordination Challenge (MACO) (Wang et al., 2022) collects the classic coordination tasks from different multi-agent learning literature and improve their difficulty. It contains 6 representative problems and can be divided into different types. First, it can be classified into factored and non-factored games which means whether an explicit decomposition of global rewards is needed. The factored games can be further divided into pairwise and non-pairwise games which represent whether the task requires pairwise or higher-order coordination. Meanwhile, the tasks can be classified into dynamic and static games according to whether the task characterizes static coordination relationships among agents. We show MACO benchmark in Table 2.

Table 2. Scenarios in multi-agent coordination benchmark.

Task	Factored	Pairwise Coordination	Dynamic Coordination	# Agents
Aloha	X	X		10
Pursuit	X	X	X	10
Hallway	X			12
Sensor	X		X	15
Gather		–		5
Disperse		–	X	12

Aloha (Hansen et al., 2004; Oliehoek, 2010) consists of 10 agents in $2 \quad 5$ array. At each time step, They need to send messages to reach the maximum backlog of 5. Message collided and needs to be resent when 2 adjacent agents send messages simultaneously. Each successful transmission will gain a positive global 0.1 reward and the collision will gain a negative global reward -10. In the beginning, each unit starts with 1 backlog. Meanwhile, a new packet will arrive probability of 0.6 if the maximum backlog has not been reached at each time step.

Pursuit (Benda, 1986; Stone & Veloso, 2000; Son et al., 2019), also known as Predator and Prey, consists of 10 agents which aim to capture 5 preys in a $10 \quad 10$ array. A prey is captured if two agents catch it simultaneously and a global reward of 10 is received. Each time step will earn a global reward -1.

Hallway (Wang et al., 2020) consists of 12 agents, and each agent randomly spawns at a state in each chain. Agents can observe their own position and choose to move left, move right, or keep still. If the agents within the same group arrive at the same state g simultaneously, a global reward is received. If more than one groups reach g, they receive a global punishment. Each agent can only observe its own position.

Sensor (Lesser et al., 2003; Zhang & Lesser, 2011) consists of 15 agents in $3 \quad 5$ array, and each agent can scan its eight neighbors with a global cost -1. If $n_{scan} > 2$ agents scan one of 3 targets which wander randomly in the gird at the same time step, a global reward $1.5 \quad n_{scan}$ is received. Each agent can observe the id and position of targets in the neighborhood.

Gather (Wei & Luke, 2016) consists of 5 agents, and each agent has 3 actions which denote 3 common goals. In each episode, one of 3 common goals is set up randomly. A higher global reward of 10 is received if all agents choose this goal, and a relatively lower reward of 5 is received if no agents choose this goal.

Disperse consists of 12 agents, and each agent has 4 actions which denote to work in one of 4 hospitals. At each time step, one hospital is chosen randomly with x agents needed. If $n_{choose} < x$ agents choose this hospital, a negative global reward $x \quad n_{choose}$ is received. Each agent can only observe the local hospital's ID and its need for the next time step.

B.2. Hyperparameter Configurations

We apply our method on 3 popular value decomposition baselines, QMIX (Rashid et al., 2018), QPLEX (Wang et al., 2021a), and RODE (Wang et al., 2021b). We implement these baselines and corresponding deconfouned training via PyMARL (Samvelyan et al., 2019). The experiment configurations are shown in Table 3. The other hyper-parameters of the baselines are set the same as that in SMAC.

Table 3. Common settings of different methods.

Settings	MACO	StarCraft II
Batch size	32	32
Replay buffer size	5000	5000
Exploration time steps	500000	50000 (500000 for super hard maps)
Start exploration rate	1	1
End exploration rate	0.05	0.05
TD-loss discount	0.9	0.9
Target central critic update interval	200 episodes	200 episodes
Evaluation interval	10000 time steps	10000 time steps
Evaluation battle number	300 episodes	32 episodes
Learning rate	0.0005	0.0005
Optimizer	RMSProp	RMSProp (Adam for Corridor and 6h_vs_8z)
Sampling Times D	4	8

C. Graph-based Methods in Multi-Agent Reinforcement Learning

There are two branches of graph-based methods in MARL. (1) The first branch (Naderializadeh et al., 2020) utilizes GNNs to estimate the joint value actions in the central mixer. Since the credits K is not influenced by the graph, it still suffers from the confounding effect. (2) The second branch (Guestrin et al., 2002; Böhmer et al., 2020; Wang et al., 2022) leverage coordination graph (CG), in which the vertices represent the agents and edges represent the payoff functions defined over the joint action-observation space of the connected agents. Meanwhile, distributed constraint optimization (DCOP) algorithm (Cheng, 2012), implemented by *Max-Plus* (Pearl, 1988; Standers et al., 2009), is used to find actions with the maximum value. Coordination graph (Guestrin et al., 2002) $G =< V, E >$ represents each vertex $v_i \in V$ as an agent i, and undirected edges $\{i,j\} \in E$ as coordination dependencies among agents. A CG induces a decomposition of the joint value function into utility functions q_i and payoff functions q_{ij}:

$$Q_{tot}(s,u) = \frac{1}{|V|}\sum_{i} q_i(u_i|s) + \frac{1}{|E|}\sum_{\{i,j\}\in E} q_{ij}(u_i,u_j|s). \tag{22}$$

The message exchanging can be computed as:

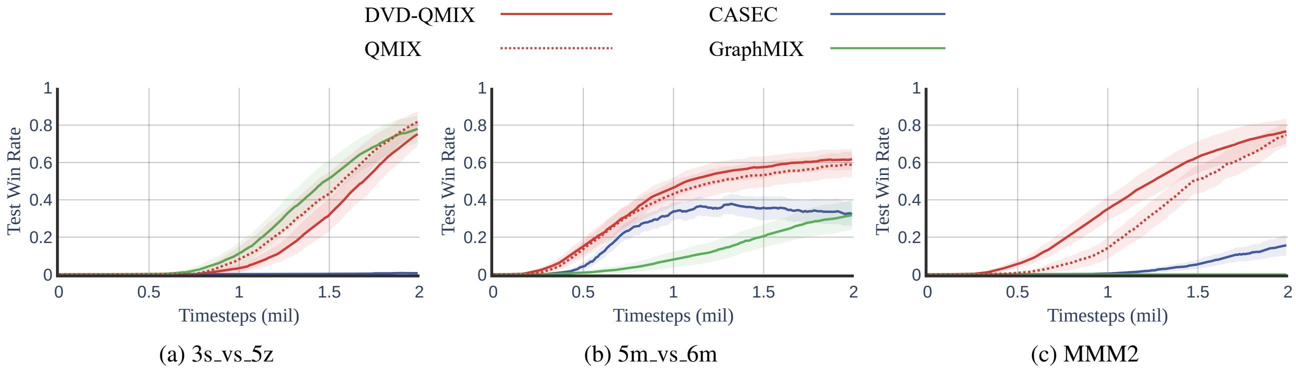
$$\mu_{ij}(u_j) \leftarrow \max_{u_i}\{\frac{1}{|V|}q_i(u_i|s) + \frac{1}{|E|}q_{ij}(u_i,u_j|s) + \sum_{\{k,i\}\in E}\mu_{ki}(u_i) - \mu_{ji}(u_i)\}. \tag{23}$$

After several iterations of message exchanging, each agent can find its optimal action by computing

$$u_i = \arg\max_{u_i}\{\frac{1}{|V|}q_i(u_i|s) + \sum_{\{k,i\}\in E}\mu_{ki}(u_i)\}. \tag{24}$$

For cyclic graphs, a normalization constant from c_{ij} each message is needed for guaranteeing the convergence (Murphy et al., 1999; Crick & Pfeffer, 2002; Yedidia et al., 2003).

As we mentioned before, we leverage a proxy confounder G to complete the backdoor adjustment, which is implemented by *trajectory graph*. Moreover, the graph attention layer (Veličković et al., 2018) is utilized to exchange messages among agents. Though part of the improvement of our method comes form graph networks, deconfounded training plays a more important role. To demonstrate the efficiency of our method, we compared DVD with the representative methods of two branches, GraphMIX (Naderializadeh et al., 2020) and CASEC (Wang et al., 2022) which are extensions of QMIX (Rashid et al., 2018). The experiments are conducted on the StarCraft II benchmark, and the results are shown in Figure 6, where the hard of the maps as well as the number of the agents increase gradually.



(a) 3s_vs_5z (b) 5m_vs_6m (c) MMM2

Figure 6. Performance comparison with graph-based methods on the StarCraft II micro management benchmark.

GraphMIX shows comparable results on map 3s_vs_5z. However, the advantage comes from QMIX itself rather than the graph structure. While in the map 5m_vs_6m and MMM2, both GraphMIX and CASEC behave poorly. The main reason is that they underestimate the role of credit assignment procedure, and message passing in the graph is not enough to guide the learning of decentralized policies. This disadvantage is especially obvious when the number of agents is large.

D. Other forms of intermediate node G

The trajectory graph is just one (efficient) way to realize the proxy confounder G. In fact, G can be replaced with the any other forms. Neither graph network nor the attention mechanism can eliminate the spurious correlation without the proposed causal theory. We perform the experiments to realize G with a simple form with just one layer neural network (DNN), the results are shown in Figure 7. We can see that a simple form also improves the baseline significantly, and a rational topological structure of the trajectory graph will further help deconfounding and improve the performance. Qatten (Yang et al., 2020b) adopts a more complicated network of attention mechanism but gains little improvement.

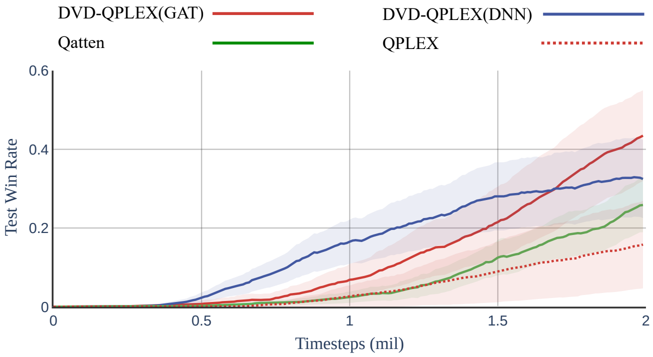


Figure 7. Performance of different form of G and attention mechanism on map MMM2 of the StarCraft II micro management benchmark.