CADP: Towards Better Centralized Learning for Decentralized Execution in MARL

Yihe Zhou[1] , **Shunyu Liu**[1*] , **Yunpeng Qing**[1] , **Tongya Zheng**[2] ,
Kaixuan Chen[3,4] , **Jie Song**[1] , **Mingli Song**[3,4]
[1]Zhejiang University
[2]Zhejiang Provincial Engineering Research Center for Real-Time SmartTech in Urban Security Governance, School of Computer and Computing Science, Hangzhou City University
[3]State Key Laboratory of Blockchain and Data Security, Zhejiang University
[4]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
{zhouyihe, liushunyu, qingyunpeng, chenkx, sjie, brooksong}@zju.edu.cn, doujiang_zheng@163.com

Abstract

Centralized Training with Decentralized Execution (CTDE) has recently emerged as a popular framework for cooperative Multi-Agent Reinforcement Learning (MARL), where agents can use additional global state information to guide training in a centralized way and make their own decisions only based on decentralized local policies. Despite the encouraging results achieved, CTDE makes an independence assumption on agent policies, which limits agents from adopting global cooperative information from each other during centralized training. Therefore, we argue that the existing CTDE framework cannot fully utilize global information for training, leading to an inefficient joint exploration and perception, which can degrade the final performance. In this paper, we introduce a novel Centralized Advising and Decentralized Pruning (CADP) framework for MARL, that not only enables an efficacious message exchange among agents during training but also guarantees the independent policies for decentralized execution. Firstly, CADP endows agents the explicit communication channel to seek and take advice from different agents for more centralized training. To further ensure the decentralized execution, we propose a smooth model pruning mechanism to progressively constrain the agent communication into a closed one without degradation in agent cooperation capability. Empirical evaluations on different benchmarks and across various MARL backbones demonstrate that the proposed framework achieves superior performance compared with the state-of-the-art counterparts. Our code is available at https://github.com/zyh1999/CADP

1 Introduction

Cooperative Multi-Agent Reinforcement Learning (MARL) has recently been attracting increasing attention from research communities, attributed to its capability on training autonomous agents to solve many real-world tasks, such as video games [Vinyals *et al.*, 2019], traffic light systems [Wu *et al.*, 2020] and smart grid control [Xu *et al.*, 2024]. However, learning cooperative policies for various complex multi-agent systems remains a major challenge. Firstly, the joint action-observation space grows exponentially with the number of agents, leading to a scalability problem when considering the multi-agent system as a single-agent one to search the optimal joint policy [Rashid *et al.*, 2018; Sunehag *et al.*, 2018]. Moreover, optimizing agent policies individually also suffers from non-stationarity due to the partial observability constraint [Hong *et al.*, 2022; Jiang *et al.*, 2024]. To tackle these problems, Centralized Training with Decentralized Execution (CTDE) is proposed as a popular learning framework for MARL [Lowe *et al.*, 2017]. In CTDE, as depicted in Figure 1(a), decentralized agent policies are trained by a centralized module with additional global state information, while agents select actions only based on their own local observation without any communication

In recent years, the CTDE framework has been widely used in MARL, including Value Decomposition (VD) methods [Sunehag *et al.*, 2018; Rashid *et al.*, 2018; Wang *et al.*, 2021; Son *et al.*, 2019; Rashid *et al.*, 2020; Liu *et al.*, 2023; Liu *et al.*, 2024; Kapoor *et al.*, 2024] and Policy Gradient (PG) methods [Lowe *et al.*, 2017; Foerster *et al.*, 2018; Yu *et al.*, 2022; Kuba *et al.*, 2022], which achieves the state-of-the-art performance in different benchmarks. Despite its promising success, we argue that the centralized training in CTDE is not centralized enough. This is to say, the existing CTDE framework cannot take full advantage of global information for centralized training. Specifically, agent policies are assumed to be independent of each other [Wang *et al.*, 2023], and the existing CTDE framework only introduces global information in the centralized module, while agents are not granted access to global information even when centralized training. This partial observability limits agents to search better global joint policy [Hong *et al.*, 2022;

[*] Corresponding author.

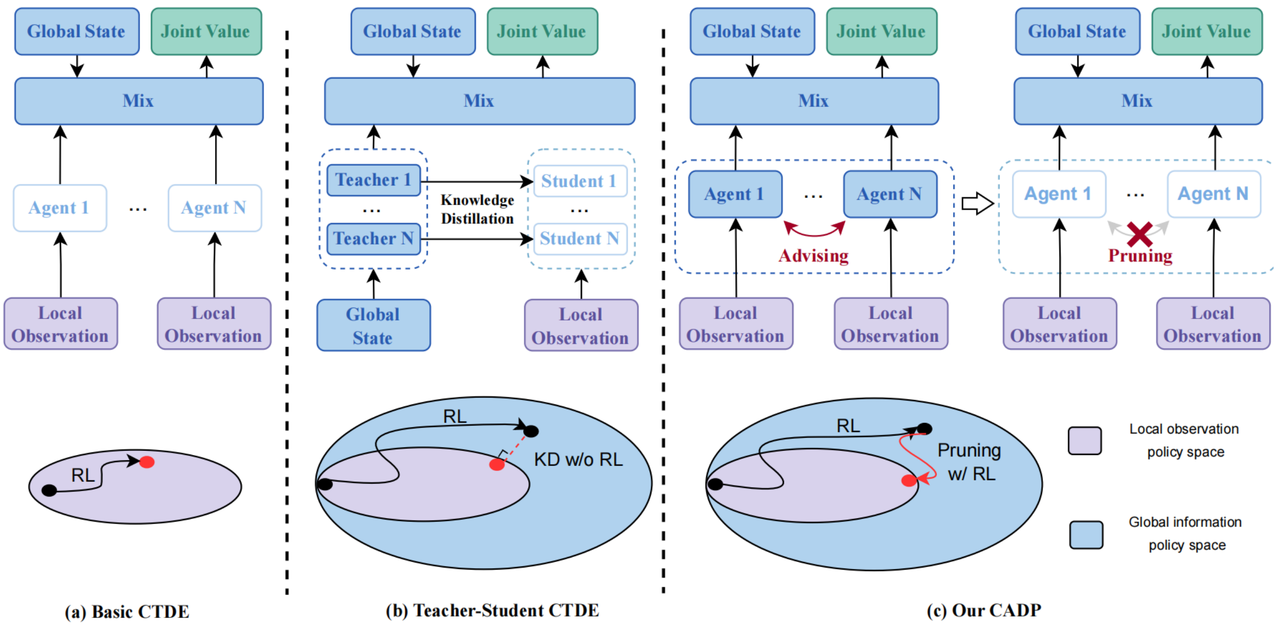


Figure 1: Comparisons between existing frameworks and our CADP. (a) Basic CTDE framework. Each agent learns its individual policy by optimizing the joint value of the centralized module with the global state. (b) Teacher-student CTDE framework. This framework introduces knowledge distillation to improve agent learning, where teachers use global information and students use local information. (c) Our CADP framework. Agents exchange their advice during centralized training then prune the dependence (still with RL) for decentralized execution.

Chen *et al.*, 2024; Zhao *et al.*, 2022].

To remedy this issue, several prior efforts propose to design a teacher-student CTDE framework [Hong *et al.*, 2022; Chen *et al.*, 2024; Zhao *et al.*, 2022], as depicted in Figure 1(b). These works enable teacher agents to use the global state information during centralized training, while student agents with local observation can imitate the behaviors from teacher agents via knowledge distillation. However, these works just simply take the additional state information as the input of agent policy, which follows the independence assumption on agent policies. Therefore, the agents still make their own decisions without considering the policies of other agents during centralized training. In this way, the expressiveness of the joint policy is inevitably limited, leading to an inefficient joint exploration and perception, which can degrade the final performance.

In this paper, we propose a novel Centralized Advising and Decentralized Pruning framework, termed as CADP, to enhance basic CTDE with global cooperative information. As depicted in Figure 1(c), CADP enables agents to exchange advice with each other instead of only using global state information during centralized training. This approach is aligned with a common way of human communication, where humans often offer helpful and tailored advice based on their knowledge and beliefs instead of simply providing their own information [Stenning *et al.*, 2006]. Therefore, the centralized advising mechanism allows agents to deliver team-oriented actions for better cooperation. To generate the final decentralized policies, we further propose to smoothly prune the dependence relationship among agents via a dedicated auxiliary loss function.

Our main contribution is the dedicated attempt that adopts agent communication to enhance basic CTDE framework for fully centralized training with decentralized execution. We propose a novel Centralized Advising and Decentralized Pruning (CADP) framework to promote explicit agent cooperation during training while still ensure the independent policies for execution. CADP is designed to provide a new general training framework for different MARL methods based on CTDE. Experiments conducted on various benchmarks show that the proposed CADP framework yields results superior to the state-of-the-art methods.

2 Related Works

Teacher-student CTDE Framework. Conventional CTDE-based methods fall short of fully utilizing global information for training, leading to an inefficient exploration and perception of the joint-policy and resulting in performance degration. Therefore, several recent works attempt to improve CTDE with the teacher-student framework. IGM-DA [Hong *et al.*, 2022] firstly proposes centralized teacher with decentralized student frameworks, where the teacher models use the global state while the student models use the partial observation. It adds an additional knowledge distillation loss to enable the teacher model to assist in training the student model. In CTDS [Zhao *et al.*, 2022], the teacher model still uses the agent observation, but the field of view is set to infinite during centralized training. PTDE [Chen *et al.*, 2024] trains a network to aggregate local observation and global state, resulting in a better representation of agent-specific global information. Most of these methods focus on distillation to transfer the knowledge from the teacher to the student for decentralized execution, which is essentially an offline imitation learning. Thus, the agents still ignore the policies of other agents and make their own decision during centralized training, resulting in an inefficient cooperative exploration.

Communication in MARL. Communication [Wang *et al.*, 2020a; Sukhbaatar *et al.*, 2016; Li *et al.*, 2023; Nayak *et al.*, 2023; Lo *et al.*, 2024] is also widely used in MARL, which facilitates information transmission between agents for effective collaboration when agents can only access their own observations. Recently, many methods have used attention as the main mechanism of communication [Das *et al.*, 2019; Jiang and Lu, 2018; Iqbal and Sha, 2019; Yuan *et al.*, 2022; Mao *et al.*, 2020b; Hu *et al.*, 2024], where self-attention can be considered as a message exchanging mechanism with other agents. The communication paradigm allows agents to communicate during both the training and execution stages. In contrast, CTDE emphasizes centralized training of agents while they execute their individual policies in a decentralized no-communication manner. Furthermore, to address communication constraints, various message pruning methods [Wang *et al.*, 2020b; Mao *et al.*, 2020a; Wang *et al.*, 2020c; Yuan *et al.*, 2022; Ding *et al.*, 2020] propose to compress and refine communication information, choose with whom to communicate. Furthermore, MACPF [Wang *et al.*, 2023] offers a method involving information transmission during training without necessitating it during execution. However, it only enables unidirectional forwarding of previous agents' actions, which does not qualify as comprehensive mutual communication.

3 Preliminary

3.1 Dec-POMDP

We consider a fully cooperative multi-agent task as the *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP), which is defined as a tuple $\langle \mathcal{A}, \mathcal{S}, \mathcal{U}, P, r, \Omega, O, \gamma \rangle$, where $\mathcal{A} = \{a_n\}_{n=1}^N$ is the set of N agents and $s \in \mathcal{S}$ is the global state of the environment. At each time step t, each agent $a_n \in \mathcal{A}$ receives an individual partial observation $o_t^n \in \Omega$ drawn from the observation function $O(s_t, a_n)$, then each agent chooses an action $u_t^n \in \mathcal{U}$ which forms joint action $\boldsymbol{u}_t \in \mathcal{U}^N$. This causes a transition to the next state s_{t+1} according to the state transition function $P(s_{t+1}|s_t, \boldsymbol{u}_t) : \mathcal{S} \times \mathcal{U}^N \times \mathcal{S} \to [0, 1]$. The reward function which is modeled as $r(s_t, \boldsymbol{u}_t) : \mathcal{S} \times \mathcal{U}^N \to \mathbb{R}$ and $\gamma \in [0, 1)$ is the discount factor. Each agent a_n has an action-observation history $\tau^n \in \mathcal{T} \equiv (\Omega \times \mathcal{U})^*$, on which it conditions a stochastic policy $\pi^n(u^n|\tau^n) : \mathcal{T} \times \mathcal{U} \to [0, 1]$. The joint action-observation history is defined as $\boldsymbol{\tau} \in \mathcal{T}^N$. In this work, the joint policy $\boldsymbol{\pi}$ is based on joint action-value function $Q_t^{tot}(s_t, \boldsymbol{u}_t) = \mathbb{E}_{s_{t+1:\infty}, \boldsymbol{u}_{t+1:\infty}} \left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} \mid s_t, \boldsymbol{u}_t \right]$. The final goal is to get the optimal policy $\boldsymbol{\pi}^*$ that maximizes the joint action value.

3.2 Value Decomposition in MARL

Value Decomposition (VD) is a useful technique in cooperative MARL to achieve effective Q-learning [Sunehag *et al.*, 2018; Rashid *et al.*, 2018; Wang *et al.*, 2021; Li *et al.*, 2021; Jiang *et al.*, 2021; Qing *et al.*, 2024]. It aims to learn a joint action-value function Q^{tot} to estimate the expected return given current global state s_t and joint action $\boldsymbol{u}_t$. To realize VD, a mix network $f(\cdot; \theta_v)$ with parameters θ_v is adopted as an approximator to estimate the joint action-value function Q^{tot}. $f(\cdot; \theta_v)$ is introduced to merge all individual values into a joint one $Q^{tot} = f(\boldsymbol{q}; \theta_v)$, where $\boldsymbol{q} = [Q^n]_{n=1}^N \in \mathbb{R}^N$ and Q^n with shared parameters θ_a is the action-value network of each agent a_n. Usually, $f(\cdot; \theta_v)$ is enforced to satisfy the Individual-Global-Max [Son *et al.*, 2019] principle. Therefore, the optimal joint action can be easily derived by independently choosing a local optimal action from each local Q-function Q^n, which enables Centralized Training and Decentralized Execution (CTDE). The learnable parameter $\theta = \{\theta_a, \theta_v\}$ can be updated by minimizing the Temporal-Difference (TD) loss as:

$$\mathcal{L}_{TD}(\theta) = \mathbb{E}_{\mathcal{D}} \left[\left(y^{tot} - Q^{tot} \right)^2 \right]. \tag{1}$$

where $\mathbb{E}[\cdot]$ denotes the expectation function, $\mathcal{D}$ is the replay buffer of the transitions, $y^{tot} = r + \gamma \hat{Q}^{tot}$ is the one-step target and $\hat{Q}^{tot}$ is the target network [Mnih *et al.*, 2015]. Additionally, owing to the fact that the partial observability often limits the agent in the acquisition of information, the agent policy usually uses past observations from history [Sunehag *et al.*, 2018].

3.3 Policy Gradient in MARL

Policy gradient (PG) [Yu *et al.*, 2022; Lowe *et al.*, 2017; Foerster *et al.*, 2018] has been proposed as a competent alternative to directly optimize the policy. In the domain of cooperative MARL, the PG mechanism complies with the CTDE constraint through the learning of an individual actor $\pi^n(u^n|\tau^n)$, and a centralized critic $V(s) : \mathcal{S} \to \mathbb{R}$, for all agents. To leverage global information during centralized training, the value functions V^n typically incorporate the global state s as input to ensure accurate estimation of the expected value. By adopting this decentralized execution approach, the consequent implicit joint policy achieves a fully independent factorization and agent policies are assumed to be independent of each other [Fu *et al.*, 2022]: $\boldsymbol{\pi}(\boldsymbol{u} \mid \boldsymbol{\tau}) = \prod_{n=1}^N \pi^n (u^n \mid \tau^n)$. PG directly maximizes the expected discounted return $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ as the objective. Thus, the loss function is defined as $\mathcal{L}_{actor} = -\mathbb{E}_{\boldsymbol{\pi}}[R_t]$. To optimize the actor π, we can perform policy gradient [Sutton and Barto, 2018] as:

$$\nabla_{\theta_{\boldsymbol{\pi}}} \mathcal{L}_{AC}(\theta_{\boldsymbol{\pi}}) = -\mathbb{E}_{s \sim p^{\boldsymbol{\pi}}, \boldsymbol{u} \sim \boldsymbol{\pi}} \left[\sum_{t=0}^{T} R_t \nabla_{\theta_{\boldsymbol{\pi}}} \log \boldsymbol{\pi} (\boldsymbol{u} \mid \boldsymbol{\tau}) \right], \tag{2}$$

where $p^{\boldsymbol{\pi}}$ is the state distribution. In particular, R_t is often replaced by $r_t + \gamma V(s_{t+1}) - V(s_t)$ which is calculated by the critic function to reduce the high variance.

4 Method

To introduce global cooperative information for agent training, we propose the Centralized Advising and Decentralized Pruning (CADP) framework, as shown in Figure 2. In general, CADP performs CTDE to enable each agent to learn its

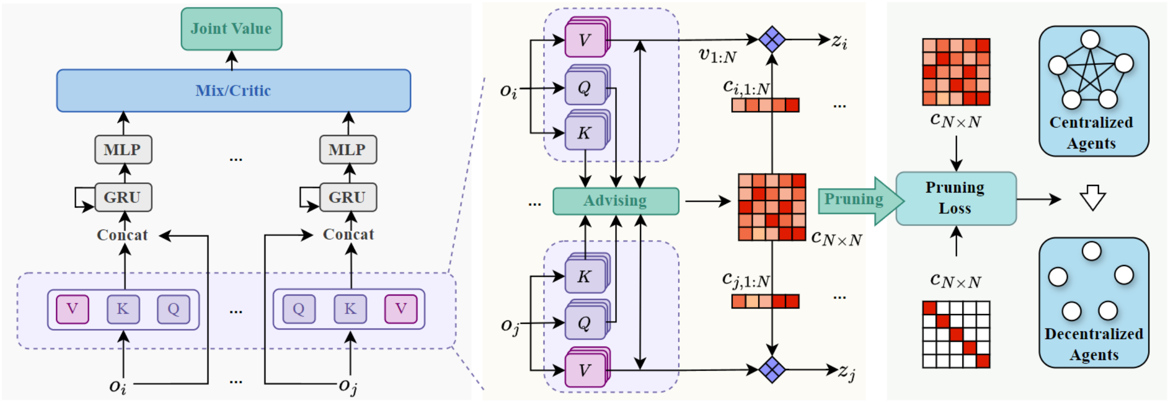


Figure 2: Illustrative diagram of the proposed Centralized Advising and Decentralized Pruning (CADP) framework. At centralized training stage, the agent model will use the Q, K, V modules, while at decentralized execution stage, the agent model only uses V module.

individual policy network. At the heart of our design is introducing the explicit cooperative information exchanging of agents for sufficient centralized training to enhance the agent policy network. To cope with the decentralized execution paradigm, we design a model self-pruning mechanism, which prompts the centralized model to evolve into a decentralized model smoothly. The overall framework is finally summarized. Besides, for the sake of clarity, we employs the VD method as an illustrative example to introduce the proposed CADP framework, while CADP is also readily applicable to the PG method.

4.1 Advice Exchanging

The widely adopted CTDE framework only introduces the global state for agents in the mix/critic module, leading to that an agent policy network only perceives its local observation instead of the global states. In contrast, we design a novel centralized training scheme to augment the agent policy network from the local information of an individual agent to the global cooperative information from all agents, inferring better action decisions.

Formally, we employ an agent's confidence c for all agents to highlight its personalized confidence weights of other agents when receiving interchangeable cooperative advice from them, where the higher confidence corresponds to the more useful information of agents. The whole process can be reduced to a self-attention mechanism [Vaswani *et al.*, 2017], where we set messages key k and value v, respectively, while confidence c is considered as the dot product of the key k of other agents and query q of itself. q, k and v are all linear projections of the local observation o. The formula is written as:

$$\alpha_{i,j} := \frac{q_i \cdot k_j}{\sqrt{d_x}}, c_{i,j} := \frac{exp(\alpha_{i,j})}{\sum_{k=1}^{N} exp(\alpha_{i,k})}, \tag{3}$$

where k_j means the message key k of agent j and q_i means the query q of agent i, d_x is the scaling coefficient and $c_{i,j}$ is the confidence from agent i to agent j. Finally, the aggregating information z_i of the agent i is obtained by taking the weighted sum of the value according to the confidence $c_{i,1:N} := (c_{i,1}, c_{i,2}, \cdots, c_{i,N})$:

$$z_i := \sum_{j=1}^{N} c_{i,j} \cdot v_j, \tag{4}$$

where v_j means the value v of agent j. Through this step, each agent refers to the cooperative information of others. Then, we combine the aggregating information z in the previous step with the agent's own local information h, and finally output the action value Q:

$$h_i^t := GRU([z_i, o_i], h_i^{t-1}), \quad Q^i := MLP(h_i^t), \tag{5}$$

where $GRU(\cdot,\cdot)$ stands for Gated Recurrent Unit. Notably, we have incorporated residual connections into the input of the GRU network. This short-circuit mechanism allows us to simultaneously leverage representations with and without advice exchanging, for enhancing training stability and performance.

Since our work focuses on the agent policy module, we can adopt different mix modules such as VDN [Sunehag *et al.*, 2018], QMIX [Rashid *et al.*, 2018] and QPLEX [Wang *et al.*, 2021] to generate Q^{tot}. Besides, if the final output of our agent module is a policy distribution π^i instead of Q^i (performing normalization after Equation 5), we can also employ MAPPO [Yu *et al.*, 2022].

4.2 Model Self-Pruning

In pursuit of facilitating decentralized execution, the current model needs to evolve into the decentralized model depending only on itself, rather than global information. In this step, we design a simple yet effective model self-pruning method to achieve this. We claim that if the following conditions are satisfied, the agent model is a decentralized model:

$$c_{i,1:N} = \mathbf{e}^i, \forall i \in [1, N], \tag{6}$$

where $\mathbf{e}^i$ means the i-th standard basis vector (an one-hot vector). In this way, we can directly apply the agent model to decentralized execution as there is no advice from other agents used. Furthermore, we just apply the value v to produce the output z without key k and other components. For the convenience of expression, we refer to the model that only requires their own values v without the self-attention mechanism as the decentralized model (CADP (D)). On the other hand, the model using self-attention mechanism is referred to as the centralized model (CADP (C)). A decentralized model actually presents that the agent's confidence $c_{i,1:N}$ of all agents is equal to the one-hot vector $\mathbf{e}^i$ in the execution stage. It is required to smoothly swap from the centralized training with exchanging confidence to the decentralized execution with the one-hot confidence $c_{i,1:N}$. Therefore, we design an auxiliary loss function named pruning loss $\mathcal{L}_p$ to help the decentralized agent gradually alleviate the dependence of other agents, which is given as:

$$\mathcal{L}_p(\theta_a) = \sum_{i=1}^{N} D_{KL}(\mathbf{e}^i \| c_{i,1:N}), \tag{7}$$

where θ_a means the parameters in the agent module. In the pruning loss, smaller $\mathcal{L}_p$ means agents rely less on the others.

4.3 Overall Framework

In our framework, we adopt advice exchanging step at agent policy module to produce more thoughtful and team-oriented action decisions. After the model achieves satisfactory performance, we start to use the pruning loss $\mathcal{L}_p$, prompting the centralized model to evolve into a decentralized one smoothly. Our mix module implementation uses QMIX [Rashid *et al.*, 2018] as a basic backbone for its robust performance and its simplicity of architecture, but it is readily applicable to the other mix/critic method since our framework focuses on the agent policy module.

To sum up, training our CADP framework contains two main loss functions. The first one is naturally the original TD loss $\mathcal{L}_{TD}$ mentioned on Equation 1, which enables each agent to learn its individual agent policy by optimizing the joint-action value of the mix module. Unlike IGM-DA [Hong *et al.*, 2022] and CTDS [Zhao *et al.*, 2022], to avoid performance degradation and reduce computing costs, our pruning loss $\mathcal{L}_p$ is not introduced at the very beginning. We add pruning loss when the centralized model reaches a high level. Therefore, the total loss of our framework is formulated as follows. (Here we take the method of value decomposition as an example. For the PG method, we just need to add the pruning loss to the loss of actor $\mathcal{L}_{AC}$ mentioned on Equation 2):

$$\mathcal{L}_{tot}(\theta) = \mathcal{L}_{TD}(\theta_{mix}, \theta_a) + \sigma(t)\mathcal{L}_p(\theta_a), \tag{8}$$

where θ_{mix} stands for the parameters of mix network in our method and t is the timestep of the training. $\sigma(\cdot)$ is a threshold function which is defined as follows:

$$\sigma(t) = \begin{cases} \alpha & if \quad t \geq T, \\ 0 & Otherwise, \end{cases} \tag{9}$$

where T is a hyperparameter and α is the coefficient for trading off loss term. From a practical application perspective, it suffices to incorporate the pruning loss after the centralized model achieves satisfactory performance in online learning. Nevertheless, in Section 5.4, we still conduct ablation experiments for T to further investigate its impact. In addition, we provide pseudocode in Appendix D.

5 Experiments

To demonstrate the effectiveness of the proposed CADP framework, we conduct experiments on the StarCraft II micromanagement challenge and Google Research Football benchmark. We aim to answer the following questions: (1) Can CADP outperforms the methods under traditional CTDE framework? (Section 5.1) (2) Can CADP outperforms the methods under teacher-student CTDE framework? (Section 5.2) (3) Can CADP perform better than the message (communication) pruning methods under the CTDE constraints? (Section 5.3) (4) Can CADP, as a universal framework, be applied to different methods and achieve improvement? (Section 5.4) (5) In comparison to traditional CTDE methods, does CADP require excessive time overhead? (Appendix C). Besides, visualization is given in Appendix E.

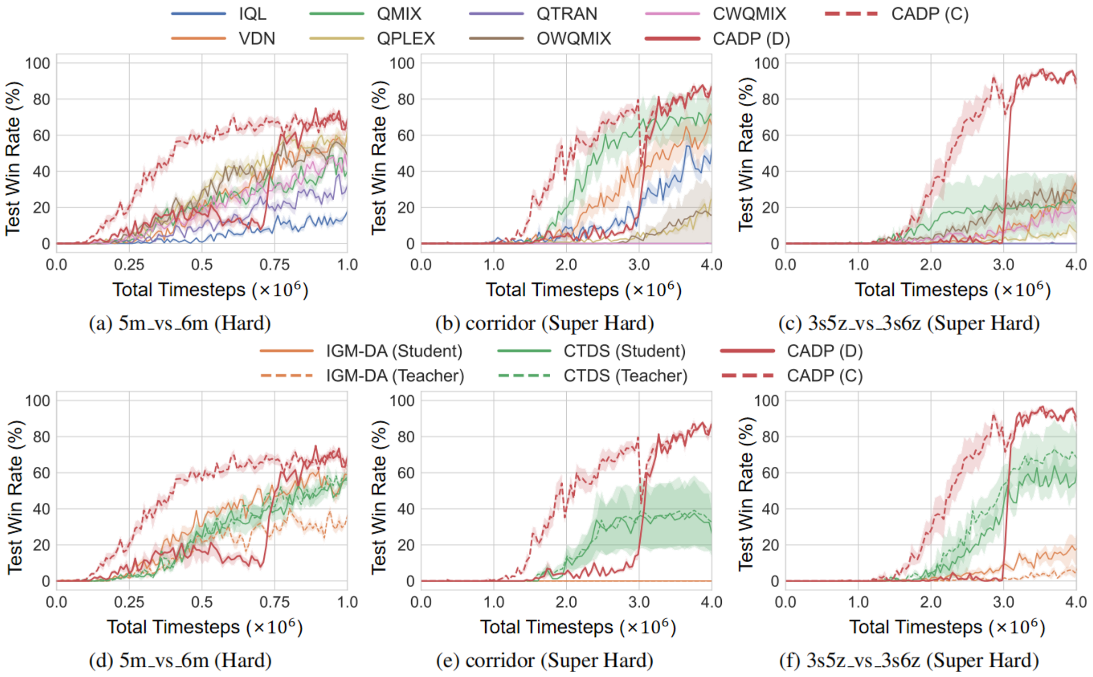
Based on these questions, we opt to compare three major categories of baseline methods: traditional CTDE-based methods, teacher-student CTDE framework methods and the message pruning methods under the CTDE constraints. Each category will be compared with CADP. Our CADP framework uses QMIX [Rashid *et al.*, 2018] as a basic backbone for its robustness and its simplicity of architecture, but it is readily applicable to the other mix/critic backbone. We show it at Section 5.4. The detailed hyperparameters are given in Appendix B , where the common hyperparameters across different methods are consistent for comparability. We also provide additional expereiments in Appendix B.

5.1 Comparison with Traditional CTDE

We select several traditional CTDE methods: VDN [Sunehag *et al.*, 2018], QMIX [Rashid *et al.*, 2018], QTRAN [Son *et al.*, 2019], QPLEX [Wang *et al.*, 2021], CWQMIX [Rashid *et al.*, 2020], OWQMIX [Rashid *et al.*, 2020] and one DTDE-based methods (regarded as a special case of CTDE): IQL [Tan, 1993] for comparison. The experimental results on different scenarios are shown in Figure 3. It can be seen that our proposed method successfully improves the final performance in the challenging tasks. Especially in the most difficult homogeneous scenario (*3s5z_vs_3s6z*) due to the different unit types, the large number of entities and the great disparity in strength between the two teams, our method can outperform baselines by a large margin.

5.2 Comparison with Teacher-Student CTDE

We also compare with the teacher-student CTDE framework methods: CTDS [Zhao *et al.*, 2022] and IGM-DA [Hong *et al.*, 2022]. The experimental results on different scenarios are shown in Figure 3. Similarly to the previous section, our method has better final performance in the challenging tasks, not only compared with the methods under teacher-student CTDE framework. Especially in the super hard scenarios (*3s5z_vs_3s6z*) and (*corridor*) due to the different unit types or the large number of entities and the great disparity in strength between the two teams, our method outperforms baselines by a large margin. In addition, after adding the pruning loss, the performance of our decentralized model rises dramatically from almost zero to near the performance of our centralized model. By contrast, in methods teacher-



(a) 5m_vs_6m (Hard)

(b) corridor (Super Hard)

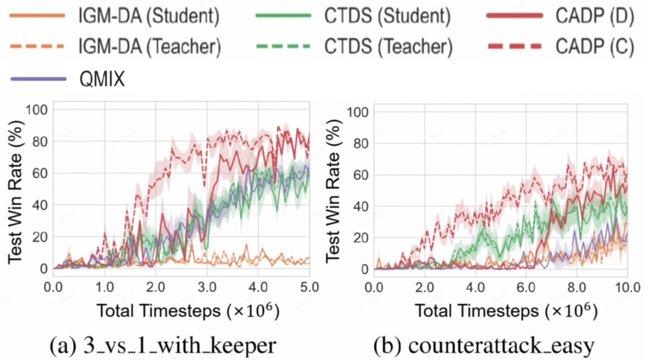
(c) 3s5z_vs_3s6z (Super Hard)

(d) 5m_vs_6m (Hard)

(e) corridor (Super Hard)

(f) 3s5z_vs_3s6z (Super Hard)

Figure 3: Learning curves of our method and baselines on the SMAC scenarios. (Upper) Comparision with the methods under the CTDE and DTDE frameworks. (Lower) Comparision with the methods under the teacher-student CTDE framework. CADP(C) means our centralized model, while CADP(D) means our decentralized model which will be used for decentralized execution.



(a) 3_vs_1_with_keeper

(b) counterattack_easy

Figure 4: Learning curves of our CADP method and baselines on the Google Research Football (GRF) scenarios.

student CTDE framwork, the performances of student models and teacher models climb together slowly and even sometimes the teacher models fail for training. Furthermore, our CADP method is the only method which can ultimately reach almost consistent performance for decentralized model and centralized model in all scenarios.

Besides, we test teacher-student CTDE framework methods on the Google Research Football (GRF) [Kurach *et al.*, 2020] benchmark. Unlike StarCraft II, individual observations in the Google Research Football (GRF) are not partial observation, which contains as much information as global state. In this case, the teacher-student CTDE framework will no longer have a significant advantage over methods under traditional CTDE framework theoretically, since the teacher model does not have more input information than the student model and has no ability to instruct the student model. The experimental results on different scenarios are shown in Figure 4. It can be seen that only our method significantly outperforms QMIX [Rashid *et al.*, 2018]. In addition, in the method of the teacher-student framework, the student model always maintains almost the same performance as the teacher model, which verifies that the teacher model does not have the ability to guide the student model in this case and also shows that using global cooperative information is more powerful than simply providing more observation information. In GRF benchmark, we set $T = 3M$ in *3_vs_1_with_keeper* scenario and $T = 6M$ in *counterattack_easy* scenario respectively. We can see an obvious improvement of our method after adding the pruning loss $\mathcal{L}_p$ at timestep T in both.

5.3 Comparison with Message (Communication) Pruning Methods under CTDE Constraint

Furthermore, we test some message pruning methods, GACML [Mao *et al.*, 2020a], NDQ [Wang *et al.*, 2020c], MAIC [Yuan *et al.*, 2022] and MACPF [Wang *et al.*, 2023] under CTDE constraint. GACML [Mao *et al.*, 2020a] and MAIC [Yuan *et al.*, 2022] are attention-based methods, while the other is not. During training, we allow them to communicate and apply message pruning, but during execution, we cut off all communication. To ensure a fair comparison, we also used QMIX [Rashid *et al.*, 2018] as the backbone for these methods. The experimental results in the table 1 show that these methods are almost not better than basic QMIX [Rashid

Method	**5m_vs_6m**	**corridor**	**3s5z_vs_3s6z**
GACML	0.46 ± 0.10	0.30 ± 0.33	0.32 ± 0.36
NDQ	0.38 ± 0.09	0.42 ± 0.21	0.00 ± 0.00
MAIC	0.28 ± 0.10	0.01 ± 0.01	0.32 ± 0.45
MACPF	0.20 ± 0.18	0.67 ± 0.39	0.16 ± 0.14
QMIX (CTDE)	0.43 ± 0.13	0.70 ± 0.35	0.24 ± 0.36
QMIX (CADP)	**0.68 ± 0.08**	**0.84 ± 0.03**	**0.93 ± 0.03**

Table 1: The comparison of test win rate between our CADP method and the message-pruning methods.

Method	5m_vs_6m	corridor	3s5z_vs_3s6z
VDN (CTDE)	0.54 ± 0.09	0.65 ± 0.32	0.25 ± 0.18
VDN (CADP)	**0.66 ± 0.07**	**0.72 ± 0.51**	**0.85 ± 0.20**
QMIX (CTDE)	0.43 ± 0.13	0.70 ± 0.35	0.24 ± 0.36
QMIX (CADP)	**0.68 ± 0.08**	**0.84 ± 0.03**	**0.93 ± 0.03**
QPLEX (CTDE)	0.57 ± 0.13	0.20 ± 0.12	0.08 ± 0.11
QPLEX (CADP)	**0.73 ± 0.04**	**0.37 ± 0.36**	**0.96 ± 0.02**
MAPPO (CTDE)	0.85± 0.07	0.96 ± 0.03	0.35 ± 0.39
MAPPO (CADP)	**0.97± 0.03**	**0.98 ± 0.02**	**0.90 ± 0.16**

Table 2: The test win rate of different MARL methods with our CADP framework and the CTDE framework.

et al., 2018] which means when communication is completely cut off, the performance of these methods is not good.

These experimental results indicate that although message pruning can reduce communication, it is not suitable for CTDE scenarios where there is no communication at all. These message pruning methods, except MACPF [Wang *et al.*, 2023] focus on compressing and refining communication information or selecting who to communicate with, which results in smaller bandwidth. As for MACPF [Wang *et al.*, 2023], the poor performance can mostly be attributed to the insufficient information exchanging, which is solely the one-hot vector representing the actions of the previous agents. In contrast, our CADP framework is designed to enhance the CTDE framework, which does not allow any communication at all during the decentralized execution phase. Our CADP framework first trains a well-performing communication model, and then gradually and dynamically distills its knowledge into a non-communication model.

5.4 Ablation Study

Different backbones. To further verify the generability of our CADP framework, we test value-based methods: VDN [Sunehag *et al.*, 2018], QMIX [Rashid *et al.*, 2018], QPLEX [Wang *et al.*, 2021] and policy-based method: MAPPO [Yu *et al.*, 2022] under the CADP framework. Experimental results in Table 2 indicate that all these methods have shown an improvement when using the CADP framework. Especially in the most difficult homogeneous scenario (*3s5z_vs_ 3s6z*), methods with our CADP framework outperform baselines by a large margin. This observation indicates the versatility and effectiveness of our proposed CADP framework in catering to various CTDE-based methods, particularly in scenarios with high levels of difficulty.

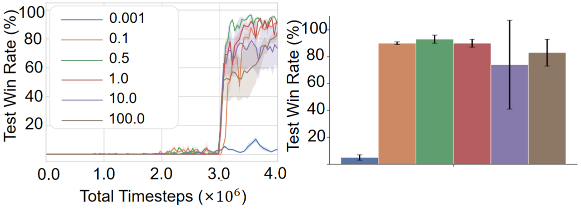


Figure 5: Ablation study on different coefficients α. The left part is the learning curves for 4M timesteps and the right part is the average test win rate of last 0.1M timesteps.

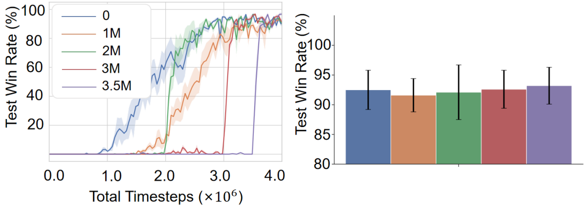


Figure 6: Ablation study on different timestep T. The left part is the learning curves for 4M timesteps and the right part is the average test win rate of last 0.1M timesteps.

Different hyperparameter. We examine the effect of the coefficient α in *3s5z_vs_3s6z* scenarios in Figure 5. We observe that there is not much difference in performance except the coefficient is too small ($\alpha = 0.001$) to affect the training. Even when we set $\alpha = 100.0$, The final performance is also not far behind the best ($\alpha = 0.5$). This means our method is very robust for the coefficient α. In addition, we consider five T values: 0, 1M, 2M , 3M, and 3.5M. Figure 6 demonstrates that using smaller T for training, the performance of decentralized starts to improve earlier but taked longer time from the beginning of improvement to convergence which may waste some computing resources while using bigger T for training, the performance of decentralized model improves more quickly but the convergence time may delay. In general, setting $T = 3M$ is a balanced choice for both sides. The experiment also shows that different values have little impact on the final performance, which may be because $\mathcal{L}_{TD}$ is always dominant in training before the convergence of the centralized model.

6 Conclusion

In this paper, we argue the traditional CTDE framework is not centralized enough, since it falls short of fully utilizing global information for training, leading to an inefficient exploration of the joint-policy, and resulting in performance degration. Thus, we propose a novel Centralized Advising and Decentralized Pruning framework, termed as CADP, to enhance basic CTDE with global cooperative information. It is noting that our focus is not to design a new communication method. Our main contribution is adopting agent communication to enhance the basic CTDE framework for fully centralized training and still guarantees the independent policy for decentralized execution. As CADP provides a light yet efficient framework, we believe there will be more discussion and exploration under CADP framework.

Acknowledgments

This work was supported in part by the CCF-Baidu Open Fund under Grant No. CCF-BAIDU OF202410, in part by the Hangzhou Joint Funds of the Zhejiang Provincial Natural Science Foundation of China under Grant No. LHZSD24F020001, in part by the Zhejiang Province High-Level Talents Special Support Program "Leading Talent of Technological Innovation of Ten-Thousands Talents Program" under Grant No. 2022R52046, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LMS25F020012, and in part by the advanced computing resources provided by the Supercomputing Center of Hangzhou City University.

References

[Chen *et al.*, 2024] Yiqun Chen, Hangyu Mao, Tianle Zhang, Shiguang Wu, Bin Zhang, Jianye Hao, Dong Li, Bin Wang, and Hongxing Chang. PTDE: personalized training with distillated execution for multi-agent reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2024.

[Das *et al.*, 2019] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, 2019.

[Ding *et al.*, 2020] Ziluo Ding, Tiejun Huang, and Zongqing Lu. Learning individually inferred communication for multi-agent cooperation. In *Annual Conference on Neural Information Processing Systems*, 2020.

[Foerster *et al.*, 2018] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*, 2018.

[Fu *et al.*, 2022] Wei Fu, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. Revisiting some common practices in cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2022.

[Hong *et al.*, 2022] Yitian Hong, Yaochu Jin, and Yang Tang. Rethinking individual global max in cooperative multi-agent reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, 2022.

[Hu *et al.*, 2024] Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Learning multi-agent communication from graph modeling perspective. In *International Conference on Learning Representations*, 2024.

[Iqbal and Sha, 2019] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2019.

[Jiang and Lu, 2018] Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Annual Conference on Neural Information Processing Systems*, 2018.

[Jiang *et al.*, 2021] Haobo Jiang, Jin Xie, and Jian Yang. Action candidate based clipped double q-learning for discrete and continuous action tasks. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

[Jiang *et al.*, 2024] Haozhe Jiang, Qiwen Cui, Zhihan Xiong, Maryam Fazel, and Simon S. Du. A black-box approach for non-stationary multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2024.

[Kapoor *et al.*, 2024] Aditya Kapoor, Benjamin Freed, Howie Choset, and Jeff Schneider. Assigning credit with partial reward decoupling in multi-agent proximal policy optimization, 2024.

[Kuba *et al.*, 2022] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

[Kurach *et al.*, 2020] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Google research football: A novel reinforcement learning environment. In *AAAI Conference on Artificial Intelligence*, 2020.

[Li *et al.*, 2021] Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021.

[Li *et al.*, 2023] Chuming Li, Jie Liu, Yinmin Zhang, Yuhong Wei, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. Ace: Cooperative multi-agent q-learning with bidirectional action-dependency. In *AAAI Conference on Artificial Intelligence*, 2023.

[Liu *et al.*, 2023] Shunyu Liu, Yihe Zhou, Jie Song, Tongya Zheng, Kaixuan Chen, Tongtian Zhu, Zunlei Feng, and Mingli Song. Contrastive identity-aware learning for multi-agent value decomposition. In *AAAI Conference on Artificial Intelligence*, pages 11595–11603, 2023.

[Liu *et al.*, 2024] Shunyu Liu, Jie Song, Yihe Zhou, Na Yu, Kaixuan Chen, Zunlei Feng, and Mingli Song. Interaction pattern disentangling for multi-agent reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Lo *et al.*, 2024] Yat Long Lo, Biswa Sengupta, Jakob Foerster, and Michael Noukhovitch. Learning to communicate using contrastive learning. In *International Conference on Learning Representations*, 2024.

[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. In *Annual Conference on Neural Information Processing Systems*, 2017.

[Mao *et al.*, 2020a] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning agent communication under limited bandwidth by message pruning. In *AAAI Conference on Artificial Intelligence*, 2020.

[Mao *et al.*, 2020b] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning multi-agent communication with double attentional deep reinforcement learning. *International Joint Conference on Autonomous Agents and Multi-agent Systems*, 2020.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Nayak *et al.*, 2023] Siddharth Nayak, Kenneth Choi, Wenqi Ding, Sydney Dolan, Karthik Gopalakrishnan, and Hamsa Balakrishnan. Scalable multi-agent reinforcement learning through intelligent information aggregation, 2023.

[Qing *et al.*, 2024] Yunpeng Qing, Shunyu Liu, Jingyuan Cong, Kaixuan Chen, Yihe Zhou, and Mingli Song. A2PO: towards effective offline reinforcement learning from an advantage-aware perspective. In *Annual Conference on Neural Information Processing Systems*, 2024.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2018.

[Rashid *et al.*, 2020] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted QMIX: expanding monotonic value function factorisation for deep multi-agent reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, 2020.

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

[Stenning *et al.*, 2006] Keith Stenning, Jo Calder, and Alex Lascarides. *Introduction to cognition and communication*. MIT Press, 2006.

[Sukhbaatar *et al.*, 2016] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *Annual Conference on Neural Information Processing Systems*, 2016.

[Sunehag *et al.*, 2018] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Joint Conference on Autonomous Agents and Multi-agent Systems*, 2018.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Tan, 1993] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *International Conference on Machine Learning*, 1993.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Annual Conference on Neural Information Processing Systems*, 2017.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[Wang *et al.*, 2020a] Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *International Conference on Machine Learning*, 2020.

[Wang *et al.*, 2020b] Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *International Conference on Machine Learning*, 2020.

[Wang *et al.*, 2020c] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*, 2020.

[Wang *et al.*, 2021] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021.

[Wang *et al.*, 2023] Jiangxing Wang, Deheng Ye, and Zongqing Lu. More centralized training, still decentralized execution: Multi-agent conditional policy factorization. In *International Conference on Learning Representations*, 2023.

[Wu *et al.*, 2020] Tong Wu, Pan Zhou, Kai Liu, Yali Yuan, Xiumin Wang, Huawei Huang, and Dapeng Oliver Wu. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(8):8243–8256, 2020.

[Xu *et al.*, 2024] Feiyang Xu, Shunyu Liu, Yunpeng Qing, Yihe Zhou, Yuwen Wang, and Mingli Song. Temporal prototype-aware learning for active voltage control on power distribution networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

[Yu *et al.*, 2022] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. In *Annual Conference on Neural Information Processing Systems*, 2022.

[Yuan *et al.*, 2022] Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. Multi-agent incentive communication via decentralized teammate modeling. In *AAAI Conference on Artificial Intelligence*, 2022.

[Zhao *et al.*, 2022] Jian Zhao, Xunhan Hu, Mingyu Yang, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. CTDS: centralized teacher with decentralized student for multi-agent reinforcement learning. *IEEE Transactions on Games*, 16(1):140–150, 2022.