Cooperative Policy Agreement: Learning Diverse Policy for Offline MARL

Yihe Zhou[1], Yuxuan Zheng[1], Yue Hu[1], Kaixuan Chen[3,4], Tongya Zheng[5,3], Jie Song[1], Mingli Song[3,4], Shunyu Liu[2*]

[1] Zhejiang University
[2] Nanyang Technological University
[3] State Key Laboratory of Blockchain and Data Security, Zhejiang University
[4] Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
[5] Big Graph Center, Hangzhou City University
{zhouyihe, zyxuan, huyue2002, chenkx, sjie, brooksong}@zju.edu.cn,
doujiang_zheng@163.com, shunyu.liu@ntu.edu.sg

Abstract

Offline Multi-Agent Reinforcement Learning (MARL) aims to learn optimal joint policies from pre-collected datasets without further interaction with the environment. Despite the encouraging results achieved so far, we identify the *policy mismatch* problem that arises from employing diverse offline MARL datasets, a highly important ingredient for cooperative generalization yet largely overlooked by existing literature. Specifically, in the case that offline datasets exhibit various optimal joint policies, policy mismatch often occurs when individual actions from different optimal joint actions are combined in a way that results in suboptimal joint actions. In this paper, we introduce a novel Cooperative Policy Agreement (CPA) method, that not only mitigates the policy mismatch problem but also learns to generate diverse joint policies. CPA firstly introduces an autoregressive decision-making mechanism among agents during offline training. This mechanism enables agents to access the actions previously taken by other agents, thereby facilitating effective joint policy matching. Moreover, diverse joint policies can be directly obtained through sequential action sampling from the autoregressive model. Then we further incorporate a policy agreement mechanism to convert these autoregressive joint policies into decentralized policies with a non-autoregressive form, while still ensuring the diversity of the generated policies. This mechanism guarantees that the proposed CPA adheres to the Centralized Training with Decentralized Execution (CTDE) constraint. Experiments conducted on various benchmarks demonstrate that CPA yields superior performance to state-of-the-art competitors.

Introduction

Offline Multi-Agent Reinforcement Learning (MARL) endeavors to leverage pre-collected datasets to develop multi-agent strategies without additional online exploration (Meng et al. 2021; Tseng et al. 2022; Yang et al. 2021; Tian et al. 2023; Matsunaga et al. 2023; Liu et al. 2023), serving as a feasible way for many real-world applications with high safety requirements or limited efficient simulators, such as autonomous driving (Li et al. 2024; Wang et al. 2021b) and smart grid control (Xu et al. 2020; Liu et al. 2024a; Xu et al. 2024). Akin to the single-agent offline learning, the application of offline MARL presents significant challenges posed by the Out-of-Distribution (OOD) problem (Yang et al. 2021; Qing et al. 2024), which stems from the distribution shift between the data inferred by the learned policy and that gathered by the behavior policy. This challenge is further exacerbated in multi-agent settings due to the combinatorial increase in joint actions and states. To address this, existing methods propose to impose proper conservative constraints over value decomposition (Yang et al. 2021; Shao et al. 2023) or cooperative policies (Matsunaga et al. 2023).

Despite the promising results achieved, another critical challenge in offline MARL that has been largely overlooked by existing works is the *policy mismatch* problem. In practice, offline data is often collected from multiple sources, resulting in diverse datasets that may contain various optimal joint policies (Formanek et al. 2023; Qing et al. 2024). Consequently, policy mismatch can occur when individual actions from different optimal joint actions are combined into suboptimal joint actions, as they may not be compatible with each other. Taking the XOR problem in Fig. 1 as an example, the two agents can only receive a reward if they each select different actions. Agent 1 and Agent 2 independently select action A or B with same probabilities, as both actions are components of optimal joint actions $[A, B]$ and $[B, A]$. Consequently, the final joint policy is uniformly distributed across the joint action space, resulting in a 50% probability of encountering the policy mismatch problem. Notably, a comparison of the middle and lower row of Fig. 1 demonstrates that the policy mismatch problem persists even with a complete dataset (*i.e.*, no OOD problem). It is also notable that the policy mismatch problem is more prevalent in offline MARL than in online MARL. In online MARL, as depicted in the upper row of Fig. 1, the replay buffer is continuously updated with data closely correlated to the current training policy, ensuring consistency. In contrast, offline MARL relies on a static dataset that does not adapt to the current training policy, as depicted in the middle and lower row of Fig. 1, potentially leading to various optimal joint actions for the same state. This static nature can thereby exacerbate the policy mismatch problem.

Moreover, the policy mismatch problem significantly hampers the agent ability to learn diverse policies in of-

*Corresponding author.

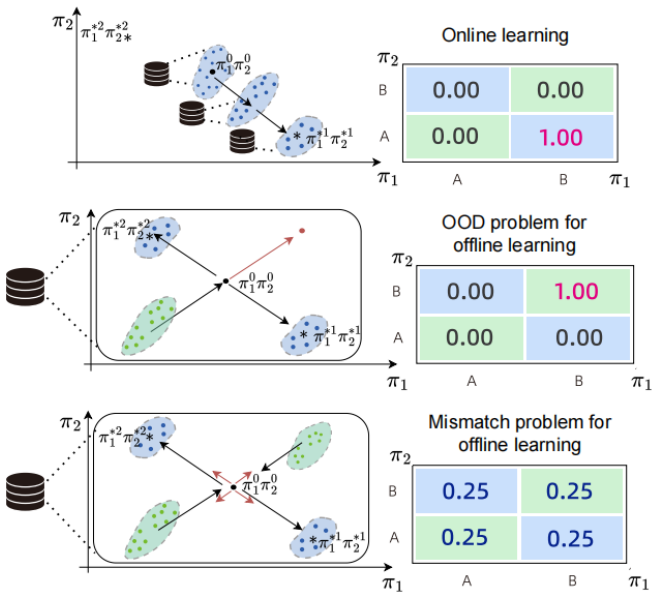


Figure 1: Illustration of the mismatch problem, considering the XOR game for example. The replay buffer (black cylinder), is continuously updated with the current policy during online learning, but remains static in offline learning. $\pi_1^{*1}\pi_2^{*1}$ and $\pi_1^{*2}\pi_2^{*2}$ represent two distinct optimal joint policies, while $\pi_1^0\pi_2^0$ represents the initial joint policy. Black arrows indicate the update direction of each iteration for the joint policy, while red arrows represent the overall update direction. The four red arrows in lower figure signify that the overall update direction is random. The green and blue blocks represent data in the replay buffer, where green denotes data with bad rewards. The three tables on the right show that the final joint policy distribution (using QMIX (Rashid et al. 2018)) under three conditions: online learning, offline learning with AB+AA+BA, and offline learning with AB+AA+BA+BB.

fline MARL. Diverse policies are highly beneficial in various MARL scenarios (Fu et al. 2022), including emergent behavior (Tang et al. 2021), cooperative exploration (Eysenbach et al. 2019; Zhou et al. 2022), learning to adapt (Balduzzi et al. 2019), and interacting with humans (Hu et al. 2020). However, this mismatch disrupts the coherence of individual actions, leading to suboptimal joint actions that fail to capture the necessary cooperative patterns required for diverse and effective multi-agent policies. Furthermore, merely addressing the policy mismatch problem might still result in convergence to a single optimal policy, thus limiting the generalization capability of agents.

In this paper, we propose a novel cooperative policy agreement method, termed as CPA, to address policy mismatch while enhancing policy diversity for offline MARL. The proposed CPA comprises two key components, namely, autoregressive decision-making mechanism and policy agreement mechanism. Firstly, the autoregressive decision-making mechanism allows agents to sequentially access the actions previously taken by other agents during offline training. This sequential action sampling ensures that agents can effectively coordinate their actions, thereby mitigating the policy mismatch problem. Moreover, CPA enables agents to generate diverse joint policies under the same state conditions through sequential action sampling from the autoregressive model. Then we further introduce a policy agreement mechanism to decentralize autoregressive joint policies while preserving their diversity. This ensures compliance with the Centralized Training with Decentralized Execution (CTDE) framework, which has become a well-established paradigm for cooperative MARL. In CTDE, agents can only make their own decisions based on decentralized local policies without any communication (Rashid et al. 2018; Hüttenrauch, Šošić, and Neumann 2017; Zhou et al. 2023; Yu et al. 2021).

Our main contributions can be summarized as follows:

- We identify and formalize the policy mismatch problem in offline MARL, a critical issue that arises from leveraging diverse datasets, to the best of our knowledge.
- We propose an autoregressive decision-making mechanism to alleviate the policy mismatch problem, while enabling agents to generate diverse cooperative policies.
- We introduce a policy agreement mechanism to transform the autoregressive joint policies into decentralized policies while preserving their diversity, ensuring compliance with the CTDE framework.
- We establish a benchmark that includes a variety of diverse datasets for offline MARL evaluation. Experimental results across various tasks demonstrate the superior performance and policy diversity of the proposed CPA.

Related Works

Multi-Agent Reinforcement Learning. Cooperative MARL focuses on enabling multiple agents to work together to achieve a common goal with shared environments and rewards. A widely used approach for cooperative MARL is centralized training with decentralized execution (CTDE) (Sunehag et al. 2018; Lowe et al. 2017; Wang et al. 2021a; Son et al. 2019; Liu et al. 2024b; Kuba et al. 2022; Luo et al. 2022), since "in many real-world settings, agents must coordinate their behavior while acting in a decentralized way. At the same time, it is often possible to train the agents in a centralized fashion in a simulated or laboratory setting, where communication constraints are lifted. (Rashid et al. 2018)" In traditional CTDE framework, methods are generally categorized into value decomposition and actor-critic methods. Value decomposition methods, such as VDN (Sunehag et al. 2018), QMIX (Rashid et al. 2018), QTRAN (Son et al. 2019), and QPLEX (Wang et al. 2021a), typically require adherence to the Individual-Global-Max (IGM) condition, which aligns individual agent decisions with global objectives. On the other hand, actor-critic methods often utilize a centralized critic to guide each agent's policy update. Prominent among these are variants of the Proximal Policy Optimization (PPO) algorithm, such as MAPPO (Yu et al. 2021), and HAPPO (Kuba et al. 2022). Additionally, there are methods

that extend beyond the traditional CTDE framework for training stability, among which a recently popular category involves sequential decision methods, such as ACE (Li et al. 2023), MAT (Wen et al. 2022), and MACPF (Wang, Ye, and Lu 2022). These methods typically incorporate the actions of the previous i-1 agents as additional inputs for the i-th agent during the execution phase, thereby enhancing inter-agent collaboration and improving team coordination.

Offline Multi-Agent Reinforcement Learning. Since some work has shown that offline MARL makes multi-agent systems more susceptible to extrapolation error (Yang et al. 2021; Fujimoto, Meger, and Precup 2019). Research has progressively begun to incorporate some multi-agent characteristics, moving beyond the straightforward application of single RL methods like BC and BCQ (Fujimoto, Meger, and Precup 2019). MAICQ (Yang et al. 2021) extends ICQ to Multi-Agent tasks with decomposed Multi-Agent joint-policy under implicit constraint. MADTKD (Tseng et al. 2022) enhances Multi-Agent Decision Transformers through a process of distilling a teacher agent with joint actions and global state, into each student agent with individual and local observation. Furthermore, AlberDICE (Matsunaga et al. 2023) introduces the DIstribution Correction Estimation (DICE) concept into the multi-agent setting to tackle the out-of-distribution (OOD) problems with joint actions.

In contrast, these methods still focus on problems about OOD or overestimation of unseen state-action pairs, which is similar to offline single RL, without giving consideration to the joint policy mismatch problem unique to offline multi-agent settings. Moreover, these methods do not focus on generating diverse joint policies from mixture datasets.

Preliminary

Multi-Agent MDP (MMDP). We consider the fully cooperative MARL setting under Multi-Agent Markov Decision Process (MMDP) (Oliehoek and Amato 2016) [1] which is defined as a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, p_0, r, \gamma \rangle$, where $\mathcal{N} = \{i\}_{i=1}^{N}$ is the set of agent indices, $s \in \mathcal{S}$ is the global state of the environment and $p_0 \in \Delta(\mathcal{S})$ is the initial state distribution. The joint action space $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$. At each time step, each agent selects an action $a_i \in \mathcal{A}_i$ via policy $\pi_i(a_i|s)$ given state s, forming a joint action $\boldsymbol{a} \in \mathcal{A}$. This causes a transition to the next state s' according to the state transition function $P(s'|s, \boldsymbol{a}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. All agents share the same reward function $r(s, \boldsymbol{a}) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $\gamma \in [0, 1)$ is the discount factor. to jointly maximize the discounted return $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$. The joint policy $\boldsymbol{\pi} = \{\pi_1, \cdots, \pi_N\}$ induces a joint action-value function $Q_t(s_t, \boldsymbol{a}_t) = \mathbb{E}_{s_{t+1:\infty}, \boldsymbol{a}_{t+1:\infty}}[R_t \mid s_t, \boldsymbol{a}_t, \boldsymbol{\pi}]$ that represents the expected discounted return under the given policy. In offline MARL, online interaction is not allowed, and the policy is optimized only using the offline dataset $\mathcal{D} = \{(s, \boldsymbol{a}, r, s')_k\}_{k=1}^{|\mathcal{D}|}$ collected by diverse data-collection agents. Besides, μ denotes the behavior policy in $\mathcal{D}$.

ICQ Method. Implicit constraint Q-learning (ICQ) is an in-sample method that updates actor and critic networks by only trusting the state-action pairs given in the dataset for more precise value estimation (Yang et al. 2021). The ICQ operator is defined as:

$$\mathcal{T}_{\text{ICQ}} Q(s, a) = r + \gamma \mathbb{E}_{a' \sim \mu} \left[\frac{1}{Z(s')} \exp\left(\frac{Q(s', a')}{\alpha} \right) Q(s', a') \right], \quad (1)$$

where $\alpha > 0$ is the Lagrangian coefficient and $Z(s) = \sum_{\tilde{a}} \mu(\tilde{a}|s) \exp\left(\frac{1}{\alpha} Q(s, \tilde{a})\right)$ is the normalizing partition function.

By extending the ICQ method to a multi-agent form, the MAICQ method trains individual policies π_i by minimizing:

$$\mathcal{L}_{\boldsymbol{\pi}}(\theta) = \sum_i \mathbb{E}_{\mathcal{D}} \left[-\frac{1}{Z^i(s)} \log(\pi_i(a_i \mid s; \theta_i)) \exp\left(\frac{w^i(s) Q^i(s, a_i)}{\alpha} \right) \right]. \quad (2)$$

As for the policy evaluation, we train $Q(s, \boldsymbol{a})$ by minimizing:

$$\mathcal{L}_Q(\phi, \psi) = \mathbb{E}_{\mathcal{D}} \left[\left(r_t + \gamma \frac{1}{Z(s'))} \exp\left(\frac{Q(s', \boldsymbol{a}')}{\alpha} \right) Q(s', \boldsymbol{a}') - Q(s, \boldsymbol{a}) \right]^2 \right]$$

$$Q(s', \boldsymbol{a}') = \sum_i w^i(s'; \psi) Q^i(s', a_i'; \phi_i) - b(s'; \psi), \quad (3)$$

where $w^i(\cdot; \psi)$ and $b(\cdot; \psi)$ are generated by the Mixer network (Rashid et al. 2018; Yang et al. 2021)

Method

In this section, we propose a new training method CPA that enables learning diverse joint cooperative policies in offline reinforcement learning while still satisfying CTDE constraints. Here we present a formal definition of the mismatch problem:

Definition 1. *For a given state s, let $\Pi(s)$ be the set of optimal joint actions, where each $(\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n) \in \Pi(s)$ represents an optimal joint action. Let $\mathcal{A}$ be the set of all possible joint actions. Mismatch(s) is the set of joint actions that are composed of individual actions from optimal joint actions in $\Pi(s)$, but which do not themselves form an optimal joint action. Formally, Mismatch(s) is written as:*

Mismatch$(s) = \{(a_1, a_2, \ldots, a_n) \in A \mid \forall i \in \{1, 2, \ldots, n\}$ $\exists(\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_{i-1}, a_i, \tilde{a}_{i+1}, \ldots, \tilde{a}_n) \in \Pi(s)$ *and* $(a_1, a_2, \ldots, a_n) \notin \Pi(s)\}$.

Autoregressive Policy

Firstly, we train agent models with autoregressive policy(Fu et al. 2022). Autoregressive policy effectively addresses the mismatch problem by allowing the i-th agent to access the action information of the previous $i-1$ agents before executing its own action. Also taking the XOR problem as an example: if agent 2 knows whether agent 1's action is A or B when it executes its own action, agent 2 can then make the correct cooperative action.

[1] We consider MMDPs rather than Dec-POMDP for simplicity.

Definition 2. *A joint policy π is defined as an **autoregressive policy** if it can be expressed as:*

$$\pi(a_1, a_2, \ldots, a_N|s) = \prod_{i=1}^{N} \pi_i(a_i \mid s, a_1, \ldots, a_{i-1}), \quad (4)$$

where each agent's policy π_i depends not only on the current state s but also on the actions of all preceding agents $(a_1, \ldots, a_{i-1})$.

Definition 3. *Let the joint policy be an **autoregressive policy**, then we extend the mismatch problem as:*

Mismatch$(s) = \{(a_1, a_2, \ldots, a_n) \in A \mid \forall i \in \{1, 2, \ldots, n\}$ $\exists(a_1, a_2, \ldots, a_{i-1}, a_i, \tilde{a}_{i+1}, \ldots, \tilde{a}_n) \in \Pi(s)$ *and* $(a_1, a_2, \ldots, a_n) \notin \Pi(s)\}$.

Proposition 1. *Mismatch(s) = $\emptyset$, if the joint policy is an **autoregressive policy**.*

The above proposition demonstrates that the use of an autoregressive policy eliminates the occurrence of the Mismatch problem, the proof of Proposition 1 is given in Appendix C. In addition, we adopt MAICQ (Yang et al. 2021) as our backbone due to its superior performance, and augment it with an autoregressive mechanism. The basic MAICQ method can be seen at Eq. (2) and Eq. (3). We introduce MAICQ with autoregressive mechanism as follows, here we use $a_{<i}$ to represent the actions of agents with indices less than i to facilitate the formulation:

$$\mathcal{L}_\pi(\theta) = \sum_i \mathbb{E}_{s,a_i\sim\mathcal{D}} \Bigg[\frac{-1}{Z^i(s, a_{<i})} \log(\pi_i(a_i \mid s, a_{<i}; \theta_i)) \exp(\frac{w^i(s) Q^i(s, a_i, a_{<i})}{\alpha}) \Bigg], \quad (5)$$

where

$$Z^i(s, a_{<i}) = \sum_{\tilde{a}_i} \mu(\tilde{a}_i \mid s, a_{<i}) \exp\left(\frac{Q^i(s', \tilde{a}_i, a_{<i}; \phi_i)}{\alpha}\right). \quad (6)$$

As for the policy evaluation, we train $Q(s, a)$ by minimizing

$$\mathcal{L}_Q(\phi, \psi) = \mathbb{E}_\mathcal{D} \Bigg[r_t + \gamma \frac{1}{Z(s')} \exp(\frac{Q\left(s', \boldsymbol{a}'\right)}{\alpha}) Q(s', \boldsymbol{a}') - Q\left(s, \boldsymbol{a}\right) \Bigg]^2. \quad (7)$$

where $Q(s', \boldsymbol{a}') = \sum_i w^i\left(s'; \psi\right) Q^i\left(s', a'_i, a'_{<i}; \phi_i\right) - b\left(s'; \psi\right)$. These two functions are corresponding to Eq. (2) and Eq. (3), Then, after optimizing the parameters for $\mathcal{L}_\pi$ and $\mathcal{L}_Q$ using the mixture offline dataset, we can get an autoregressive joint policy with many different strategies even when facing same state s. Here, we set a threshold β to control the diversity of the policy.

$$\pi_i^\beta(a|s, a_{<i}) = \frac{\exp\left(\pi_i(a|s, a_{<i}) \cdot \mathbb{I}[\pi_i(a|s, a_{<i}) > \beta]\right)}{\sum_{a' \in \mathcal{A}_i} \exp(\pi(a'|s, a_{<i}) \cdot \mathbb{I}[\pi_i(a'|s, a_{<i}) > \beta]}, \quad (8)$$

where $\mathbb{I}[\cdot]$ is an indicator function. Specifically, if the denominator is 0, we set the maximum element of $\pi_i(a|s, a_{<i})$ to 1 and all other elements to 0, which means that π_i^β becomes a deterministic policy.

Policy Agreement

Although autoregressive policies can address the mismatch problem in multi-agent offline reinforcement learning and learn diverse cooperative strategies, they fail to satisfy the CTDE constraint, because the autoregressive mechanism is essentially an expensive communication mechanism which needs $O(N)$ time cost, and is contradictory to decentralized execution (DE). To tackle this problem, we propose the Individual Policy Agreement mechanism, enabling agents to have diverse cooperative strategies even in CTDE setting. To accomplish this, we need to utilize a pre-trained agent model with an autoregressive policy as the teacher model to distill knowledge to the Individual Policy model. Specifically, we firstly obtain the set of "sample-label" pairs $\{(s, \hat{\boldsymbol{a}})_i\}$ required for knowledge distillation based on the pre-trained agent and the dataset, where state s belongs to the dataset D, and joint action $\hat{\boldsymbol{a}}$ is generated by the pre-trained agent based on s like Eq. (8).

State-Conditional Action Pool Next, we obtain the joint action pool corresponding to each state, which helps us generate diverse joint actions. The state s is processed through a Multi-Layer Perceptron $AP(\cdot; \omega_{AP})$ to generate an action pool, where $AP(\cdot; \omega_{AP}) \in \mathbb{R}^{c \times d}$. Each d-dimensional vector e_i in the action pool $AP(s; \omega_{AP})$ represents the latent encoding of a joint action. In addition, we require an encoder and decoder for the joint actions. To train this state-conditional action pool, we employ a vector quantization mechanism (van den Oord, Vinyals, and Kavukcuoglu 2017), with the loss function as follows:

$$e = \underset{e_i \in AP(s;\omega_{AP})}{\arg\min} \|z_e(\hat{\boldsymbol{a}}; \omega_e) - e_i\|_2, \quad (9)$$

$$\mathcal{L}_{VQ}(\omega_{AP}, \omega_e) = \|sg[z_e(\hat{\boldsymbol{a}}; \omega_e)] - e\|_2^2 + \beta\|sg[e] - z_e(\hat{\boldsymbol{a}}; \omega_e)\|_2^2, \quad (10)$$

where $z_e(\cdot; \omega_e)$is an encoder network, e is the vector in the action pool output by the network $AP(s; \omega_{AP})$ that is the closest match to $z_e(\hat{\boldsymbol{a}}; \omega_e)$, and $sg[\cdot]$ denotes the stop-gradient operation. The first term ensures that the encoder output is close to the nearest embedding vector, while the second term updates the embeddings to be closer to the encoder outputs.

Additionally, to train the encoder and decoder, we incorporate a reconstruction loss in the form of negative log-likelihood to ensure the decoder accurately outputs the joint action a. Since we are training individual agents, the log likelihood of the joint action for reconstruction loss can be written as:

$$e' = e - sg[(z_e(\hat{\boldsymbol{a}}; \omega_e)) - e], \quad (11)$$

$$\mathcal{L}_{R1}(\omega_e, \omega_p) = -\log(p(\hat{\boldsymbol{a}})) = -\sum_{i=1}^{N} \log(p_i(\hat{a}_i|e'; \omega_{p_i})), \quad (12)$$

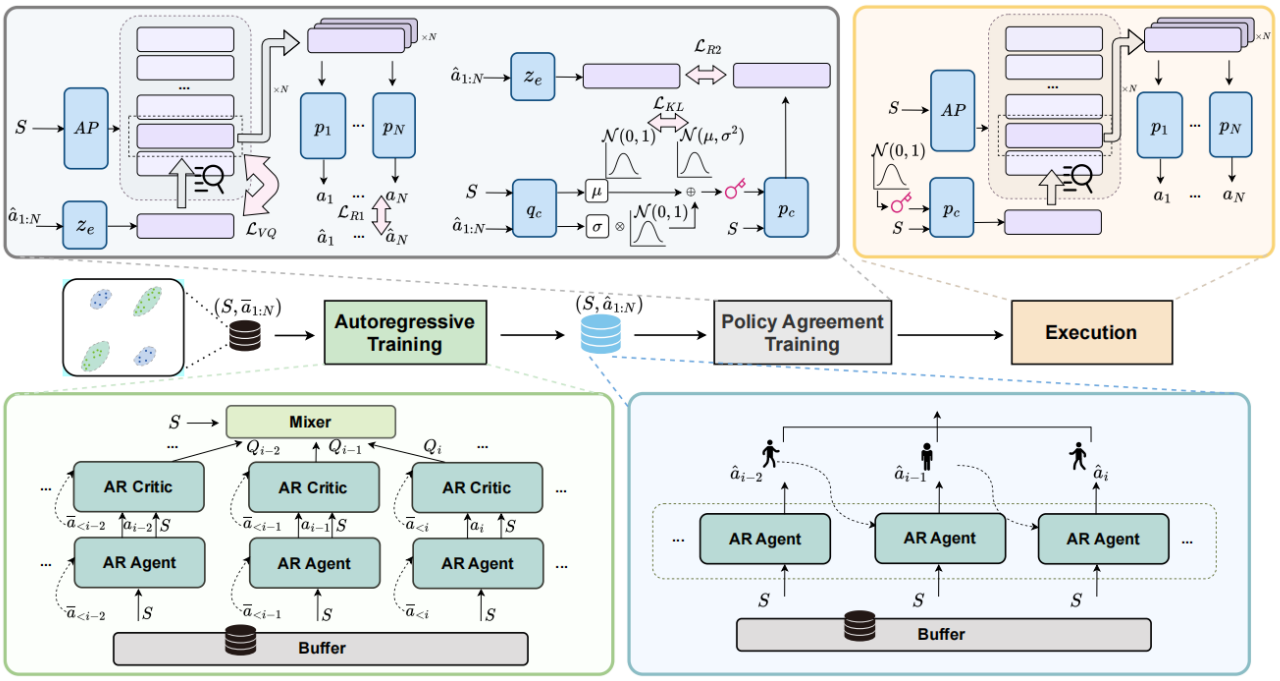


Figure 2: An illustrative diagram of the proposed CPA framework.

where $p_i(a_i; \omega_{p_i})$ is the probability distribution of action for agent i by the decoder $p_i(\cdot; \omega_{p_i})$. The total action pool loss $\mathcal{L}_{AP}$ is written as:

$$\mathcal{L}_{AP}(\omega_{AP}, \omega_e, \omega_p) = \mathcal{L}_{R1}(\omega_e, \omega_p) + \mathcal{L}_{VQ}(\omega_{AP}, \omega_e). \tag{13}$$

Action Selector with Policy Agreement After obtaining the state-conditional action pool AP through training, we need to ensure that each individual agent selects the same AP_i at the same time. This requires achieving policy agreement among different agents. To accomplish this, we introduce an agreement key k as a latent variable and employ a mechanism similar to conditional variational autoencoders (CVAE) (Sohn, Lee, and Yan 2015) to reach this goal.

The encoder of the CVAE maps the embedding vectors e and the state s to the agreement key k (latent variable):

$$q_c(k|e, s) = \mathcal{N}(k; \mu(e, s; \rho_e)), \sigma^2(e, s; \rho_e))), \tag{14}$$

where $\mu(\cdot, s; \rho_e)$ and $\sigma^2(\cdot, s; \rho_e)$ are the mean and variance functions for encoder network parameterized by ρ_e, conditioned on both the embedding vectors e and the state s. The decoder $(\mu(\cdot, s; \rho_d), \sigma^2(\cdot, s; \rho_d))$ then reconstructs the embedding vectors e from the latent variable z and the state s:

$$p_c(e|k, s) = \mathcal{N}(e; \mu(k, s; \rho_d), \sigma^2(k, s; \rho_d)). \tag{15}$$

Overall, the CVAE loss function consists of two parts: the reconstruction loss and the KL divergence. The reconstruction loss ensures that the decoder can accurately reconstruct the embedding vectors e from k and s, and is defined as:

$$\mathcal{L}_{R2}(\rho_e, \rho_d) = \mathbb{E}_{q_c(k|e,s)}[-\log p_c(e|k, s)]. \tag{16}$$

The KL divergence ensures that the learned latent variable k follows the prior distribution p_c, which is usually a standard normal distribution:

$$\mathcal{L}_{KL}(\rho_e) = D_{KL}(q_c(k|e, s) \| p_c(k)). \tag{17}$$

The total CVAE loss $\mathcal{L}_C$ is then written as:

$$\mathcal{L}_C(\rho_e, \rho_d) = \mathcal{L}_{R2}(\rho_e, \rho_d) + \mathcal{L}_{KL}(\rho_e). \tag{18}$$

By minimizing this loss, we can learn a latent representation that serves as the agreement key k, ensuring that all agents select the same joint action from the action pool AP given the state s.

After these training steps, when individual agents need to perform decentralized execution, each agent needs to copy the shared AP network as well as their respective p_i networks. Then, they need to pre-set a common key k to ensure policy agreement. Notably, setting a common key does not require explicit communication; there are several methods to achieve this: (1) Set the same random seed and a consistent random generation mechanism before execution. (2) Use the current time as the random seed to generate the common key.

Overall Framework

Our framework consists of two main parts. Firstly, we pretrain agents policies with autoregressive mechanism. Then, we transfer the ability of these agents to generate diverse policies to non-autoregressive agents with policy agreement. Initially, we train a "state-conditional action pool" to discretize the possible effective joint actions corresponding to each state. Subsequently, we use a CVAE method to model the distribution of the embedding frequencies within the action pool. The latent variable k serves as a common key among all agents to ensure "policy agreement".

Experiments

We aim to answer the following main questions: (1) Can autoregressive policy (AR) alleviate the mismatch problem? (Tab. 1) (2) Can AR policy learn diverse cooperative strategies from mixture datasets? (Fig. 3, Fig. 3 and Fig. 5) (3) Can policy agreement (PA), a non-AR mechanism learn diverse cooperative strategies from AR policy? (Table 1, Fig. 3, Fig. 4 and Fig. 5)

Our baselines include: Behavioral Cloning (BC), BCQ (Fujimoto, Meger, and Precup 2019), CQL (Kumar et al. 2020), MAICQ (Yang et al. 2021), and AlberDICE (Matsunaga et al. 2023). The detailed hyperparameters are given in Appendix B, where the common training parameters across different methods are consistent to ensure comparability.

Benchmarks

To demonstrate the effectiveness of the proposed methods, we conduct experiments on a series of classic coordination benchmarks, the stateless scenarios: XOR (Matsunaga et al. 2023; Fu et al. 2022) and Permutation (Fu et al. 2022); the stateful scenarios: Bridge (Matsunaga et al. 2023), Sensor (Zhang and Lesser 2011; Wang et al. 2022), and Aloha (Hansen, Bernstein, and Zilberstein 2004; Wang et al. 2022). Further details of the scenarios and the datasets can be found in the Appendix D.

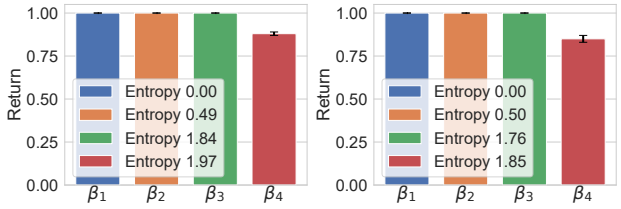
Deterministic Policy

Since most offline multi-agent reinforcement learning methods do not support the generation of effective diverse joint actions, we first conduct a deterministic policy comparison to examine whether our method addresses the mismatch problem for experimental fairness. It is noteworthy that although our method supports the generation of multiple valid joint actions, by selecting the action corresponding to the highest logits output of the autoregressive model and setting the action pool size to 1, both AR and PA can be made to output deterministic policies.

The results on different datasets of varied quality are shown in Tab. 1. Compared with the state-of-the-art baseline methods, our proposed methods, both Autoregressive method (AR) method and Policy Agreement (PA) method successfully improve the final performance in all datasets.

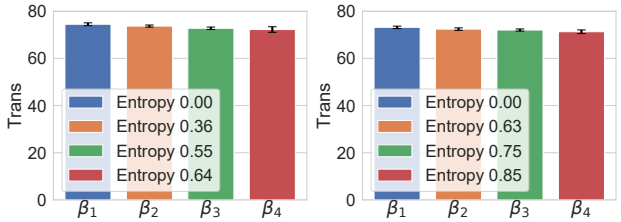
Notably, many methods exhibit a lack of correlation between dataset quality and final performance on certain datasets. For instance, methods such as BC and BCQ in Bridge, BCQ, CQL, AlberDICE in Sensor, and BC, BCQ, CQL, MAICQ and AlberDICE in Aloha demonstrate this issue severely. This indicates that the problem of mismatch, where "combining individual actions from different optimal joint actions can result in a suboptimal joint action," is prevalent in these methods. In other words, due to improper matching of individual actions, agents fail to learn relatively better joint actions from higher quality datasets. In contrast, our methods, including both AR and PA, largely avoid this issue, suggesting better resilience to the mismatch problem.

Furthermore, we can see that the PA method performs nearly identically to the AR method in the deterministic policy setting. This indicates we have successfully incorporated policy agreement mechanism to convert autoregressive joint policies into decentralized, non-autoregressive policies.



(a) AR (b) PA

Figure 3: Diversity-performance bar charts for AR and PA in Permutation scenario. We vary β to change the diversity of policy. $\beta_1 = 1$, $\beta_2 = 1/(0.8n)$, $\beta_3 = 1/n$ and $\beta_4 = 1/(1.5n)$, where $n = 4$ represents the size of the individual action space. The range of the entropy is $[0, \log_2 n] = [0, 2]$.



(a) AR (b) PA

Figure 4: Diversity-performance bar charts for AR and PA in Aloha scenario. We vary β to change the diversity of policy. $\beta_1 = 1$, $\beta_2 = 1/(1.5n)$, $\beta_3 = 1/(2n)$ and $\beta_4 = 1/(3n)$, where $n = 2$ represents the size of the individual action space. The range of the entropy is $[0, \log_2 n] = [0, 1]$.

Diverse Policy

To evaluate our method's capability to produce diverse joint actions, we construct diversity-performance bar charts by varying threshold β. These bar charts demonstrate how our methods, including both AR and PA, maintain performance as the diversity of the agent policy increases. Here we choose Permutation and Aloha scenarios (poor datasets) for evaluation. Additionally, since most methods performed poorly in the aforementioned comparisons of Deterministic Policies, rendering further comparisons of Diverse Policies impractical. Nevertheless, we still provide more diversity-performance experiments in the Appendix.

To provide a formal description of the diversity of agent policy, we introduce the concept of policy average entropy.

$$\mathbb{E}[H] = \mathbb{E}_{s\sim\boldsymbol{\pi}}\left[\frac{1}{N}\sum_{i=1}^{N}(-\sum_{a_i} P(a_i|s)\log_2 P(a_i|s))\right]. \tag{19}$$

For an autoregressive agent, the average entropy of policy can be defined as:

$$P(a_i|s) = \sum_{a_{i-1}}\cdots\sum_{a_1}\prod_{j=1}^{i}\pi_j(a_j|s, a_{<j}). \tag{20}$$

Env	Dataset	BC	BCQ	CQL	MAICQ	AlberDICE	AR (ours)	PA (ours)
XOR	AB+BA	0.80 ± 0.40	0.00 ± 0.00	0.40 ± 0.49	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	AA+AB+BA	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Complete	0.40 ± 0.49	0.00 ± 0.00	0.60 ± 0.49	0.40 ± 0.49	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Per	Poor	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Medium	1.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Good	0.40 ± 0.49	0.00 ± 0.00	0.20 ± 0.40	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Bridge	Poor	-17.57 ± 7.85	-11.54 ± 3.34	-18.52 ± 1.54	-12.80 ± 5.35	-1.40 ± 0.26	-1.27 ± 0.02	-1.27 ± 0.02
	Medium	-9.04 ± 9.54	-15.41 ± 2.30	-15.62 ± 2.48	-5.17 ± 3.48	-1.29 ± 0.00	-1.24 ± 0.00	-1.24 ± 0.00
	Good	-4.03 ± 5.50	-16.23 ± 1.08	-4.00 ± 5.52	-3.13 ± 3.77	-1.27 ± 0.03	-1.24 ± 0.00	-1.24 ± 0.00
Sensor	Poor	15.91 ± 1.75	4.43 ± 1.77	0.54 ± 0.75	0.00 ± 0.00	15.23 ± 0.79	28.86 ± 0.19	28.39 ± 0.72
	Medium	20.20 ± 5.38	6.22 ± 7.21	0.12 ± 0.23	0.00 ± 0.00	23.83 ± 0.23	28.95 ± 0.62	28.71 ± 0.32
	Good	28.32 ± 0.28	4.62 ± 2.79	12.24 ± 5.92	0.00 ± 0.00	17.00 ± 1.66	29.67 ± 0.35	29.22 ± 0.72
Aloha	Poor	2.25 ± 1.30	1.80 ± 1.94	31.15 ± 9.33	20.20 ± 11.43	49.24 ± 0.55	74.43 ± 0.60	73.12 ± 0.51
	Medium	1.52 ± 0.13	2.50 ± 4.00	13.53 ± 7.21	4.71 ± 1.90	49.42 ± 0.90	74.85 ± 0.31	73.25 ± 0.63
	Good	1.61 ± 0.11	0.00 ± 0.00	14.19 ± 3.58	0.99 ± 0.50	48.31 ± 1.69	76.12 ± 0.63	75.34 ± 0.84

Table 1: Mean performance and standard error (over 5 random seeds) with deterministic policies on different datasets.

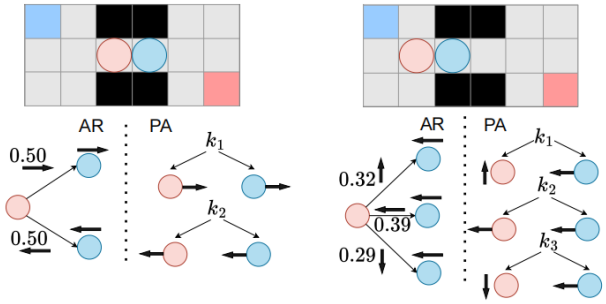


Figure 5: Visualization of learned policies at the initial state and at a subsequent state after one step, respectively.

On the other hand, for policy agreement agent, the average entropy of policy can be defined as:

$$P(a_i|s) = \int_{\mathbb{R}^n} f(k)\pi_i(a_i|s,k)dk, \tag{21}$$

where $f(\cdot)$ denotes the probability density function of the standard normal distribution. Here, we employ Monte Carlo sampling to approximate the average entropy for both.

The experiments in Fig. 3 and Fig. 4 demonstrate that our method can produce diverse and effective joint actions when facing the same state s. Furthermore, Fig. 3 and Fig. 4 also indicate that the performance of our methods decreases very slowly as diversity increases, demonstrating its ability to produce diverse effective joint policies. Additionally, the performance of our PA method, without autoregressive mechanism, is comparable to that of our AR method. This suggests that our designed Policy Agreement mechanism successfully learns diverse joint polices from the AR model.

Visualization

We select Bridge scenario for a simplified visualization: Fig. 5 demonstrates how the autoregressive model and the non-autoregressive model achieve effective collaboration through their respective mechanisms. Firstly, the autoregressive model (AR) can adjust its current actions based on the varying previous actions of agents. On the other hand, the non-autoregressive model employs a PA mechanism, utilizing the common agreement key to generate individual actions that correspond to the same joint action. Both mechanisms effectively avoid the mismatch problem.

Conclusion

This paper argues that the significant challenge in offline multi-agent reinforcement learning is not only OOD problem but also mismatch problem. By introducing an autoregressive decision-making mechanism, our method ensures proper action matching among agents during training, thereby mitigating the mismatch problem. Additionally, our method leverages mixture datasets to train agents capable of generating multiple effective joint actions, enhancing the diversity and robustness and of agent strategies. Furthermore, by incorporating the policy agreement mechanism, our method eliminates the need for communication.

Limitations and Future Works. The limitation of CPA method is its focus on the MMDP setting, neglecting the difficulties faced by agents in Dec-POMDP setting. As a result, when the discrepancy between partial observations and the global state information is significant, our policy agreement mechanism may not work effectively under CTDE constraints. In such cases, exchanging observations may be needed to acquire state information. But even with the need to exchange observations, policy agreement mechanism still represents a significant improvement over autoregressive mechanism, since the communication overhead for autoregressive mechanism is $O(N)$, while for exchanging observations is $O(1)$. Future work will explore how to apply policy agreement mechanisms within Dec-POMDP settings.

Acknowledgments

This work was supported in part by the Hangzhou Joint Funds of the Zhejiang Provincial Natural Science Foundation of China under Grant No. LHZSD24F020001, in part by the Fundamental Research Funds for the Central Universities under Grant No. 226-2024-00058, and 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.

References

Balduzzi, D.; Garnelo, M.; Bachrach, Y.; Czarnecki, W.; Pérolat, J.; Jaderberg, M.; and Graepel, T. 2019. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is All You Need: Learning Skills without a Reward Function. In *International Conference on Learning Representations*.

Formanek, C.; Jeewa, A.; Shock, J.; and Pretorius, A. 2023. Off-the-Grid MARL: Datasets with Baselines for Offline Multi-Agent Reinforcement Learning. In *International Joint Conference on Autonomous Agents and Multi- agent Systems*.

Fu, W.; Yu, C.; Xu, Z.; Yang, J.; and Wu, Y. 2022. Revisiting Some Common Practices in Cooperative Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning*.

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*.

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic Programming for Partially Observable Stochastic Games. In *AAAI Conference on Artificial Intelligence*.

Hu, H.; Lerer, A.; Peysakhovich, A.; and Foerster, J. N. 2020. "Other-Play" for Zero-Shot Coordination. In *International Conference on Machine Learning*.

Hüttenrauch, M.; Šošić, A.; and Neumann, G. 2017. Guided Deep Reinforcement Learning for Swarm Systems. *arXiv preprint arXiv:1709.06011*.

Kuba, J. G.; Chen, R.; Wen, M.; Wen, Y.; Sun, F.; Wang, J.; and Yang, Y. 2022. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. In *International Conference on Learning Representations*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. In *Annual Conference on Neural Information Processing Systems*.

Li, C.; Liu, J.; Zhang, Y.; Wei, Y.; Niu, Y.; Yang, Y.; Liu, Y.; and Ouyang, W. 2023. ACE: Cooperative Multi-Agent Q-learning with Bidirectional Action-Dependency. In *AAAI Conference on Artificial Intelligence*.

Li, Z.; Wang, Q.; Wang, J.; and He, Z. 2024. A Flexible Cooperative MARL Method for Efficient Passage of an Emergency CAV in Mixed Traffic. *IEEE Transactions on Intelligent Transportation Systems*.

Liu, S.; Luo, W.; Zhou, Y.; Chen, K.; Zhang, Q.; Xu, H.; Guo, Q.; and Song, M. 2024a. Transmission Interface Power Flow Adjustment: A Deep Reinforcement Learning Approach Based on Multi-Task Attribution Map. *IEEE Transactions on Power Systems*.

Liu, S.; Song, J.; Zhou, Y.; Yu, N.; Chen, K.; Feng, Z.; and Song, M. 2024b. Interaction Pattern Disentangling for Multi-Agent Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, S.; Zhou, Y.; Song, J.; Zheng, T.; Chen, K.; Zhu, T.; Feng, Z.; and Song, M. 2023. Contrastive Identity-Aware Learning for Multi-Agent Value Decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Annual Conference on Neural Information Processing Systems*.

Luo, S.; Li, Y.; Li, J.; Kuang, K.; Liu, F.; Shao, Y.; and Wu, C. 2022. S2rl: Do we really need to perceive all states in deep multi-agent reinforcement learning? In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1183–1191.

Matsunaga, D. E.; Lee, J.; Yoon, J.; Leonardos, S.; Abbeel, P.; and Kim, K. 2023. AlberDICE: Addressing Out-Of-Distribution Joint Actions in Offline Multi-Agent RL via Alternating Stationary Distribution Correction Estimation. In *Annual Conference on Neural Information Processing Systems*.

Meng, L.; Wen, M.; Yang, Y.; Le, C.; Li, X.; Zhang, W.; Wen, Y.; Zhang, H.; Wang, J.; and Xu, B. 2021. Offline Pre-trained Multi-Agent Decision Transformer: One Big Sequence Model Tackles All SMAC Tasks. *arXiv preprint arXiv:2112.02845*.

Oliehoek, F. A.; and Amato, C. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer Publishing Company, Incorporated, 1st edition. ISBN 3319289276.

Qing, Y.; Liu, S.; Cong, J.; Chen, K.; Zhou, Y.; and Song, M. 2024. Advantage-Aware Policy Optimization for Offline Reinforcement Learning. *arXiv preprint arXiv:2403.07262*.

Rashid, T.; Samvelyan, M.; de Witt, C. S.; Farquhar, G.; Foerster, J. N.; and Whiteson, S. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning*.

Shao, J.; Qu, Y.; Chen, C.; Zhang, H.; and Ji, X. 2023. Counterfactual Conservative Q Learning for Offline Multi-agent Reinforcement Learning. In *Annual Conference on Neural Information Processing Systems*.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Annual Conference on Neural Information Processing Systems*.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D.; and Yi, Y. 2019. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning*.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *International Joint Conference on Autonomous Agents and Multi-agent Systems*.

Tang, Z.; Yu, C.; Chen, B.; Xu, H.; Wang, X.; Fang, F.; Du, S. S.; Wang, Y.; and Wu, Y. 2021. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. In *International Conference on Learning Representations*.

Tian, Q.; Kuang, K.; Liu, F.; and Wang, B. 2023. Learning from Good Trajectories in Offline Multi-Agent Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.

Tseng, W. C.; Wang, T. H. J.; Lin, Y. C.; and Isola, P. 2022. Offline Multi-Agent Reinforcement Learning with Knowledge Distillation. In *Annual Conference on Neural Information Processing Systems*.

van den Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural Discrete Representation Learning. In *Annual Conference on Neural Information Processing Systems*.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2021a. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations*.

Wang, J.; Xu, W.; Gu, Y.; Song, W.; and Green, T. 2021b. Multi-Agent Reinforcement Learning for Active Voltage Control on Power Distribution Networks. In *Annual Conference on Neural Information Processing Systems*.

Wang, J.; Ye, D.; and Lu, Z. 2022. More Centralized Training, Still Decentralized Execution: Multi-Agent Conditional Policy Factorization. *arXiv preprint arXiv:2209.12681*.

Wang, T.; Zeng, L.; Dong, W.; Yang, Q.; Yu, Y.; and Zhang, C. 2022. Context-Aware Sparse Deep Coordination Graphs. In *International Conference on Learning Representations*.

Wen, M.; Kuba, J. G.; Lin, R.; Zhang, W.; Wen, Y.; Wang, J.; and Yang, Y. 2022. Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. In *Annual Conference on Neural Information Processing Systems*.

Xu, F.; Liu, S.; Qing, Y.; Zhou, Y.; Wang, Y.; and Song, M. 2024. Temporal Prototype-Aware Learning for Active Voltage Control on Power Distribution Networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Xu, X.; Jia, Y.; Xu, Y.; Xu, Z.; Chai, S.; and Lai, C. S. 2020. A multi-agent reinforcement learning-based data-driven method for home energy management. *IEEE Transactions on Smart Grid*, 11(4): 3201–3211.

Yang, Y.; Ma, X.; Li, C.; Zheng, Z.; Zhang, Q.; Huang, G.; Yang, J.; and Zhao, Q. 2021. Believe What You See: Implicit Constraint Approach for Offline Multi-Agent Reinforcement Learning. In *Annual Conference on Neural Information Processing Systems*.

Yu, C.; Velu, A.; Vinitsky, E.; Wang, Y.; Bayen, A.; and Wu, Y. 2021. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*.

Zhang, C.; and Lesser, V. R. 2011. Coordinated Multi-Agent Reinforcement Learning in Networked Distributed POMDPs. In Burgard, W.; and Roth, D., eds., *AAAI Conference on Artificial Intelligence*.

Zhou, Y.; Liu, S.; Qing, Y.; Chen, K.; Zheng, T.; Huang, Y.; Song, J.; and Song, M. 2023. Is Centralized Training with Decentralized Execution Framework Centralized Enough for MARL? *arXiv preprint arXiv:2305.17352*.

Zhou, Z.; Fu, W.; Zhang, B.; and Wu, Y. 2022. Continuously Discovering Novel Strategies via Reward-Switching Policy Optimization. In *International Conference on Learning Representations*.