Aligning Credit for Multi-Agent Cooperation via Model-based Counterfactual Imagination

Jiajun Chai
Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
chaijiajun2020@ia.ac.cn

Yuqian Fu
Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
fuyuqian2022@ia.ac.cn

Dongbin Zhao
Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
dongbin.zhao@ia.ac.cn

Yuanheng Zhu
Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
yuanheng.zhu@ia.ac.cn

ABSTRACT

Recent years have witnessed considerable progress in model-based reinforcement learning research. Inspired by the significant improvement in sample efficiency, researchers have explored its application in multi-agent scenarios to mitigate the huge demands in training data of multi-agent reinforcement learning (MARL) approaches. However, existing methods retain the training framework designed for single-agent settings, resulting in inadequate promotion of multi-agent cooperation. In this work, we propose a novel model-based MARL method called Multi-Agent Counterfactual Dreamer (MACD). MACD introduces a centralized imagination with decentralized execution (CIDE) framework to generate higher-quality pseudo data for policy learning, thus further improving the algorithm's sample efficiency. Moreover, we address the credit assignment and non-stationary challenges by performing an additional counterfactual trajectory based on the learned world model. We provide a theoretical proof that this counterfactual policy update rule maximizes the multi-agent learning objective. Empirical studies validate the superiority of our method in terms of sample efficiency, training stability, and final cooperation performance when compared with several state-of-the-art model-free and model-based MARL algorithms. Ablation studies and visualization demonstration further underscore the significance of both the CIDE framework and the counterfactual module in our approach.

KEYWORDS

Model-based reinforcement learning; Multi-agent reinforcement learning; Credit assignment; Counterfactual advantage

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

ACM Reference Format:
Jiajun Chai, Yuqian Fu, Dongbin Zhao, and Yuanheng Zhu. 2024. Aligning Credit for Multi-Agent Cooperation via Model-based Counterfactual Imagination. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has received a significant amount of attention to address the multi-agent cooperation challenge [13, 22]. Numerous algorithms have been proposed to solve intrinsic problems of multi-agent learning, including credit assignment [17], non-stationary training [41], and, notably, the huge demand for training data due to the exponential growth in the joint state-action space [6, 43]. Model-based RL has been proven to be a powerful tool for enhancing the sample efficiency [9]. A world model is established to serve as a digital replica of the environment, thereby augmenting the pseudo training data through imagination rollouts [9, 30]. However, the main efforts in this direction have been paid on single-agent settings, making MARL lags thus far. Attempts have been made to integrate model-based techniques with MARL algorithms [6, 35], exemplified by the incorporation of Dreamer V2 [10] into multi-agent cooperative scenarios. However, these studies employ a fully decentralized world model to reconstruct the agents' local observation transition process, resulting in imprecise pseudo data to describe the global system state transition. Furthermore, previous methods just utilize the world model as a pseudo data sampler to enhance sample efficiency, without leveraging it to address the credit assignment and non-stationary challenges in MARL training.

To address the aforementioned challenges, lots of model-free MARL approaches have been proposed. The credit assignment problem requires methods to allocate the global reward appropriately according to agents' contributions. Implicit algorithms adopt the value factorization to allocate contributions among agents during joint training [3, 8]. Other approaches use a counterfactual module or Shapley theory to allocate the global rewards explicitly via state value functions [16, 31]. However, since the value functions used to

assign credit can only maintain accuracy around the current policy distribution, these model-free methods fail in accurate contribution evaluation. In contrast, the world model in model-based RL can provide an accurate prediction capability with a wider range. Besides, since an agent needs to evaluate the impact of the other agents' policy changes on its own dynamics, the emergence of the non-stationary issue undermines the training stability. Some methods treat the system as a unified agent during training to ensure a stationary training process [22, 27]. Communication [4, 19], sequential update [15, 33], and opponent analysis [28, 44] are also used to stabilize the training process. However, these methods still exhibit shortcomings in training efficiency, as they require interaction with the true environment to estimate the impacts of policy changes.

In this paper, our objective is to incorporate the model-based techniques to address inherent challenges in MARL, including sample efficiency, credit assignment, and non-stationary challenges. To this end, we introduce MACD, a novel model-based MARL approach, to provide a more stable and more efficient training framework. (1) We propose a centralized imagination with decentralized execution (CIDE) framework to improve the sample efficiency by producing higher-quality pseudo data. We also incorporate techniques from the state-of-the-art (SOTA) Dreamer V3 [11] algorithm to build our world model. (2) We propose a model-based counterfactual module to generate counterfactual imagination trajectories to evaluate an agent's contribution. A theoretical analysis is provided to prove that this counterfactual policy update rule maximizes the multi-agent learning objective, addressing the credit assignment and non-stationary challenges. (3) We compare MACD with several SOTA model-free and model-based MARL algorithms. Empirical results demonstrate that MACD outperforms baselines in terms of sample efficiency, training stability, and final cooperation performance. Ablation results underscore the significance of both the CIDE framework and the counterfactual module, and the long-term prediction demonstration visualizes the reconstruction and prediction accuracy of our CIDE framework.

2 PRELIMINARIES

2.1 Problem Formulation

The cooperative multi-agent task can be considered as a partially observable stochastic game with a team reward. It can be defined as a tuple $\mathscr{U} = \{\mathcal{N}, \gamma, \mathbb{S}, \mathbb{O}, \mathbb{A}, \mathbb{T}, \mathbb{R}\}$, where $\mathcal{N}$ is the agent set, and $n = |\mathcal{N}|$ is the number of agents. γ is the discount factor, $\mathbb{S}$ is the global state space, and $\mathbb{O} = \{O_i\}_{i=1}^n$ is the joint observation space, with O_i being the observation space of agent i. Similarly, $\mathbb{A} = \{A_i\}_{i=1}^n$ is the joint action space, with A_i being the action space of agent i. $\mathbb{T}(s_{t+1}|s_t, \boldsymbol{a}_t)$ is the transition function, and $\mathbb{R}$ is the reward function. The system receives a team reward $r_{t+1} = \mathbb{R}(s_t, \boldsymbol{a}_t)$ based on changes in the global state. The state value function $V^{\boldsymbol{\pi}}(s)$ of the system is defined as the discounted accumulated return under state s, and the Q function adds the joint action $\boldsymbol{a}$ as a condition:

$$\begin{aligned} V^{\boldsymbol{\pi}}(s) &= \mathbb{E}_{\boldsymbol{\pi}}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s] \\ Q^{\boldsymbol{\pi}}(s, \boldsymbol{a}) &= \mathbb{E}_{\boldsymbol{\pi}}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, \boldsymbol{a}_t = \boldsymbol{a}] \end{aligned} \tag{1}$$

Besides, the advantage function $A^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = Q^{\boldsymbol{\pi}}(s, \boldsymbol{a}) - V^{\boldsymbol{\pi}}(s)$ can evaluate the joint action $\boldsymbol{a}$ with lower variance.

2.2 Explicit Credit Assignment Techniques

Current MARL methods with explicit credit assignment leverage counterfactual value or Shapley theory [25] to directly allocate team rewards according to their estimation of agents' contributions, thereby transforming the training process into policy updates for each individual agent. For instance, difference rewards assert that team rewards can be allocated using a shaped reward formula: $\mathbb{R}(s_t, \boldsymbol{a}_t) - \mathbb{R}(s_t, \{d, \boldsymbol{a}_{-i,t}\})$, where $-i$ represents the other agents excluding agent i, and d denotes a default action like a zero vector. Similarly, COMA [7] takes this idea to marginalize agent i's action according to its policy to calculate a counterfactual advantage value:

$$A_i^{\text{COMA}}(s, \boldsymbol{a}) = Q_i^{\boldsymbol{\pi}}(s, \boldsymbol{a}) - \sum_{a_i'} \pi_i(a_i'|h_i) Q^{\boldsymbol{\pi}}(s, \{a_i', \boldsymbol{a}_{-i}\}) \tag{2}$$

where h_i is the observation-action history of agent i. At timestep t, $h_{i,t} = \{o_{i,0}, a_{i,0}, ..., o_{i,t-1}, a_{i,t-1}, o_{i,t}\}$. Some methods employ similar techniques to assess agent contributions through the marginalization of agent actions [18, 42].

Shapley value methods aim to assess individual agents' contributions by determining their marginal impact on the overall return across all agent coalitions [31, 32]. For any agent coalition denoted as $c \in \mathcal{N}$, its Q function can be expressed as the discounted cumulative return, assuming that agent actions not present in the coalition are replaced with default actions:

$$Q^{\boldsymbol{\pi}}(s, \boldsymbol{a}; c) = \mathbb{E}_{\boldsymbol{\pi}}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, \boldsymbol{a} = \{a_j\}_{j \in c} \cup \{a_\ell = d\}_{\ell \in \mathcal{N} \setminus c}] \tag{3}$$

The advantage values calculated by the Shapley theory is calculated as follows:

$$A_i^{\text{Shapley}}(s_t, \boldsymbol{a}_t) = \sum_{c \in \mathcal{N} \setminus \{i\}} \frac{|c|!(n - |c| - 1)!}{n!} \text{SV}_i^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t; c) \tag{4}$$

where $\text{SV}_i^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t; c) = Q^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t; \{c, i\}) - Q^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t; c)$. Although the Shapley value has been widely applied in the cooperative game theory, their demand in calculating for all possible coalitions makes it difficult to apply to complex multi-agent tasks.

2.3 Single- and Multi-Agent Dreamer Framework

Dreamer series algorithms [9–11] are the SOTA model-based methods. They adopt the Recurrent State-Space Model (RSSM) model to establish an efficient world model, which follows an encoder-decoder structure.

$$\text{RSSM} = \begin{cases} \text{Recurrent model:} & h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \\ \text{Representation model:} & z_t \sim q_\phi(z_t|h_t, o_t) \\ \text{Transition predictor:} & \hat{z}_t \sim p_\phi(\hat{z}_t|h_t) \\ \text{Observation predictor:} & \hat{o}_t \sim p_\phi(\hat{o}_t|h_t, z_t) \\ \text{Reward predictor:} & \hat{r}_t \sim p_\phi(\hat{r}_t|h_t, z_t) \\ \text{Discount predictor:} & \hat{\gamma}_t \sim p_\phi(\hat{\gamma}_t|h_t, z_t) \end{cases}$$

The fundamental idea of the RSSM is to convert the state transition within the original state space into that within the latent

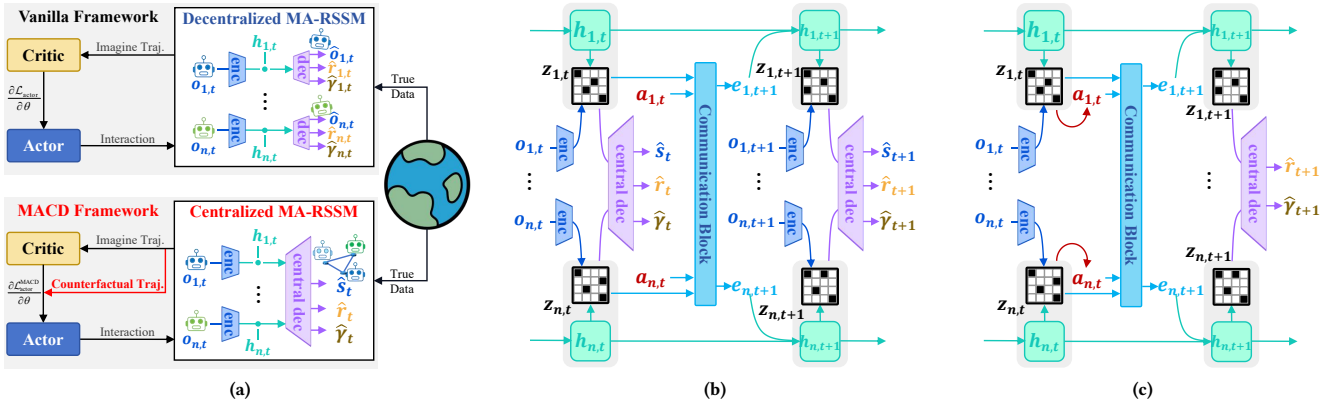


Figure 1: The framework of MA-RSSM. (a) Differences between MACD and vanilla framework. (b) Centralized imagination world model, in which the agent reconstructs the transitions on the whole system. (c) Centralized rollout within the imagination space. The communication block aggregates inputs from all agents to generate $e_{i,t}$ for agent i.

state space. The agent employs a recurrent model to produce the hidden state h_t, taking into account the previous latent state z_{t-1}, previous action a_{t-1}, and historical information from the previous hidden state h_{t-1}. This hidden state serves as input for the transition predictor, which generates the prior latent state $\hat{z}_t$. Additionally, the posterior latent state is derived from the representation model, adjusting the prior latent state based on o_t to correct for deviations during interactions with the true environment. The latent states are trained to reconstruct observations, rewards, and terminal signals, thereby enhancing their representational capacity. The agent's policy is optimized using the pseudo data generated through interactions with the world model.

MAMBA is the first Dreamer-style method in MARL [6]. It extends the RSSM from the single-agent setting to the multi-agent setting, alongside the utilization of the Proximal Policy Optimization (PPO) [24] algorithm for policy optimization. MAMBA introduces a fully-decentralized world model to adapt to decision-making and training within multi-agent environments. As shown in the vanilla framework of Figure 1 (a), each agent maintains an individualized local world model, enabling decentralized rollouts based on latent states generated by these models. Although an agent can access other agents' information through a communication block, its world model only replicates the transition of its local observations. This might result in the latent states containing only local observation information, potentially affecting algorithm performance.

3 METHOD

In this section, we introduce Multi-Agent Counterfactual Dreamer (MACD), a novel model-based MARL algorithm by introducing a *centralized imagination with decentralized execution (CIDE)* framework and a *model-based counterfactual module*. Figure 1 (a) shows the differences between MACD and vanilla framework. During the world model learning phase, our CIDE framework reconstructs global variables in a centralized manner, thereby assisting the algorithm in generating higher-quality pseudo data to facilitate subsequent policy learning. During the policy learning phase, the counterfactual module addresses the credit assignment and non-stationary problem in MARL based on the counterfactual trajectory generated by the learned world model, improving training stability and final cooperation performance.

3.1 Multi-Agent World Model Learning for Centralized Imagination

We follow the Dreamer series to establish a world model to replicate the environment into an imagination space. As shown in Figure 1 (a), the vanilla framework employed by MAMBA [6] and its modification [35] only reconstructs the transition process of agent's local observations. Consequently, the agent's latent state contains solely local information, thereby leading to instability in the policy training process. In this section, we introduce a centralized imagination world model to generate pseudo data of higher quality.

Multi-Agent Recurrent State-Space Model (MA-RSSM) for Centralized Imagination. In multi-agent settings, an agent cannot deduce the complete system's state transitions solely from its own local information. To this end, we utilize a communication block $g_\phi(\cdot)$ to consolidate the local information of all agents, producing a communication feature $\boldsymbol{e}_t$ to enrich its informational content. We present our centralized imagination world model as follows:

$$\text{MA-RSSM} = \begin{cases} \text{Communication block:} & \boldsymbol{e}_t = g_\phi(z_{t-1}, \boldsymbol{a}_{t-1}) \\ \text{Recurrent model:} & h_{i,t} = f_\phi(h_{i,t-1}, e_{i,t}) \\ \text{Representation model:} & z_{i,t} \sim q_\phi(z_{i,t}|h_{i,t}, o_{i,t}) \\ \text{Transition predictor:} & \hat{z}_{i,t} \sim p_\phi(\hat{z}_{i,t}|h_{i,t}) \end{cases}$$

$$\begin{aligned} &\text{State predictor:} && \hat{s}_t \sim p_\phi(\hat{s}_t|\boldsymbol{h}_t, \boldsymbol{z}_t) \\ &\text{Reward predictor:} && \hat{r}_t \sim p_\phi(\hat{r}_t|\boldsymbol{h}_t, \boldsymbol{z}_t) \\ &\text{Discount predictor:} && \hat{\gamma}_t \sim p_\phi(\hat{\gamma}_t|\boldsymbol{h}_t, \boldsymbol{z}_t) \end{aligned}$$

The variables' meanings in MA-RSSM are identical to those in RSSM, except for the addition of the subscript "i" to denote the agent i. The MA-RSSM part of our centralized imagination world model is the same as that of the decentralized world model. It can provide agent-specific latent states for scalable agent decision-making.

In contrast to the fully-decentralized world model, we aim to reconstruct the transition on the whole system from the perspective of all agents, thus providing centralized imagination capability for the world model. To this end, we present the centralized predictors, as shown in Figure 1 (b), to serve as decoders in the world model, thus further improving the quality of pseudo data generated within the imagination space. We utilize system-wide states $[h_{i,t}, z_{i,t}]_{i=1}^{n}$ to reconstruct the global states s_t, team rewards r_t, and continuation flags γ_t in a centralized manner. The continuation flag denotes episode termination and serves as an additional discount in the computation of accumulated discounted rewards. By reconstructing these global variables, this approach guarantees that the information embedded within the latent state space comprehensively supports the modeling of the system's state transitions.

Loss function. The multi-agent world model should use the posterior latent state generated by the representation model to reconstruct global variables while minimizing the difference between the representation model and the transition predictor. The loss function in Dreamer V3 is improved by emphasizing the update of the posterior distribution. We follow the suggestion in Dreamer V3 [11] to present the final loss function as follows:

$$\mathcal{L}(\phi) = \mathcal{L}_{\text{pred}}(\phi) + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}}(\phi) + \beta_{\text{rep}}\mathcal{L}_{\text{rep}}(\phi) \tag{5}$$

where $\beta_{\text{dyn}} = 0.5$ and $\beta_{\text{rep}} = 0.1$. The sub loss functions are:

$$\begin{aligned}
\mathcal{L}_{\text{pred}}(\phi) &= -\ln p_\phi(s_t|\boldsymbol{x}_t) - \ln p_\phi(r_t|\boldsymbol{x}_t) - \ln p_\phi(\gamma_t|\boldsymbol{x}_t) \\
\mathcal{L}_{\text{dyn}}(\phi) &= D_{\text{KL}}[\text{sg}(q_\phi(z_{i,t}|h_{i,t}, o_{i,t}))||p_\phi(\hat{z}_{i,t}|h_{i,t})] \\
\mathcal{L}_{\text{rep}}(\phi) &= D_{\text{KL}}[\text{sg}(p_\phi(\hat{z}_{i,t}|h_{i,t})||q_\phi(z_{i,t}|h_{i,t}, o_{i,t}))]
\end{aligned} \tag{6}$$

where $\boldsymbol{x}_t = \{x_{i,t}\}_{i=1}^{n}$ and $x_{i,t} = [h_{i,t}, z_{i,t}]$ are the model states for simplification, $\mathcal{L}_{\text{pred}}(\phi)$ is the prediction loss aimed at enhancing the accuracy of the predictors, $\mathcal{L}_{\text{dyn}}(\phi)$ and $\mathcal{L}_{\text{rep}}(\phi)$ represent the KL divergence losses designed to reduce the distance between the prior and posterior latent state distributions. This reduction minimizes the difference between pseudo data generated from the imagination space and actual data obtained from the true environment. The stop gradient operator is denoted as $\text{sg}(\cdot)$.

Twohot symlog prediction. Given that reward prediction involves scalar regression, conventional network architectures may encounter difficulties. Following the approach of Dreamer V3 [11], we adopt the twohot symlog module as the final output head for the reward predictor. This module employs two D-dimensional vectors to represent a scalar value. The first vector $\vec{b}$ encompasses values within a linear space range. The second vector is derived from a scalar y through the following twohot encoding procedure [14]:

$$\text{twohot}(y)_j = \begin{cases} |\vec{b}_{k+1} - y|/|\vec{b}_{k+1} - \vec{b}_k| & \text{if } j = k \\ |\vec{b}_k - y|/|\vec{b}_{k+1} - \vec{b}_k| & \text{if } j = k+1 \\ 0 & \text{else} \end{cases} \tag{7}$$

where k is the index where y locates in vector b, the subscripts denote the index of vectors. Conversely, given a vector $\vec{\ell}$, a scalar can be computed by $\text{symexp}(\vec{\ell}^T \cdot \vec{b})$. In the world model learning, the ground-truth reward is firstly transformed by $\text{symlog}(\cdot)$ function to shrink its value range and then encoded as a vector via $\text{twohot}(\text{symlog}(r_t))$. Hence, the probability likelihood in Eq. (5) can be calculated using the reward predictor's output vector.

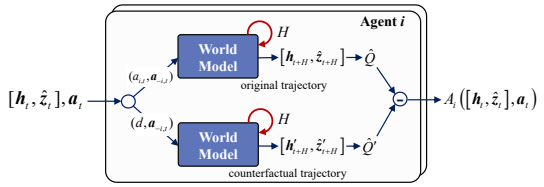


Figure 2: The counterfactual advantage module employed to assess the agent's contribution to the multi-agent system.

Network structure. We utilize the Transformer architecture [29] for the communication block and centralized predictors, while incorporating the GRU cell [5] to implement the recurrent nature to the MA-RSSM. We also follow the suggestion of Dreamer V3 to use SiLU [1] as the activation function of networks and add layer normalization between each two layers.

3.2 Counterfactual Credit Assignment for Policy Learning

In this work, we use the agent-wise model states $a_{i,t} \sim \pi_i(\cdot|x_{i,t})$ to make decisions for each agent to implement the *decentralized execution*. We utilize the learned world model to train agent policies in the policy learning phase. As denoted in Figure 1 (c), the world model is used as a data sampler to generate pseudo data for MARL training. The start points of one rollout should use the observation $\boldsymbol{o}_t$ from true environments to generate posterior latent states. Then, the transition predictor is used to generate pseudo data recurrently without accessing the true environment. The rewards and continuation flags are predicted via joint model states $\boldsymbol{x}_t$ in a centralized manner. These pseudo data are used to optimize the policies.

In the critic learning phase, since the centralized imagination leads to the same state value estimation for all agents, we use the joint model states to estimate the state value, leading to the critic network $V(\boldsymbol{x}_t; \psi)$. The critic also uses the twohot symlog module as the final output head, leading to the following critic loss function:

$$\mathcal{L}_{\text{critic}}(\psi) = -\mathbb{E}_t\left[\ln V(G_t|\boldsymbol{x}_t; \psi)\right] \tag{8}$$

where $G_t = \sum_{k=0}^{H-1} \gamma^k \hat{r}_{t+k+1} + \gamma^H V(\boldsymbol{x}_{t+H})$ is the discounted accumulated reward calculated by the pseudo data, H is the imagination horizon, and ψ is the parameter of the critic.

In the actor learning phase, we utilize the world model to perform an additional counterfactual trajectory in the imagination space to evaluate agents' contributions. Figure 2 shows the proposed model-based counterfactual module built upon the world model to address credit assignment. It evaluates the contribution of agent i under $(\boldsymbol{x}_t, \boldsymbol{a}_t)$ via the following counterfactual advantage $A_{i,t}^{\text{MACD}}$:

$$\begin{aligned}
A_{i,t}^{\text{MACD}} &= \hat{Q}_{\boldsymbol{\pi}}(\boldsymbol{x}_t, \boldsymbol{a}_t; H_c) - \hat{Q}_{\boldsymbol{\pi}}(\boldsymbol{x}_t, \{d, \boldsymbol{a}_{-i,t}\}; H_c) \\
\hat{Q}_{\boldsymbol{\pi}}(\boldsymbol{x}_t, \boldsymbol{a}_t; H_c) &= \sum_{k=0}^{H_c-1} \gamma^k \hat{r}'_{t+k+1} + \gamma^{H_c} V^{\boldsymbol{\pi}}(\boldsymbol{x}'_{t+H_c})
\end{aligned} \tag{9}$$

where $\hat{Q}_{\boldsymbol{\pi}}(\boldsymbol{x}_t, \boldsymbol{a}_t; H_c)$, $\hat{r}'_t$, and $\boldsymbol{x}'_t$ are the estimated Q values, predicted rewards, and model states via an additional rollout. This rollout starts from $\boldsymbol{x}_t$, and the first action executed in the imagination space is $\boldsymbol{a}_t$. Then, the transition predictor is used to generate pseudo data recurrently by executing joint policy $\boldsymbol{\pi}$ for H_c steps,

meaning that the actions of all agents are generated by the joint policy $\boldsymbol{\pi}$ within the counterfactual rollout. The calculation process of $\hat{Q}_{\boldsymbol{\pi}}(\boldsymbol{x}_t, \{d, \boldsymbol{a}_{-i,t}\}; H_c)$ is similar, except that the starting action for agent i is the default action d, which in practice is usually set to zero. By masking an agent's action and assessing the resulting impact on the discounted accumulated reward, the counterfactual module evaluates agent i's contribution for credit assignment. In contrast to model-free explicit MARL methods [7, 39], we employ a world model derived from supervised training to eliminate problems like overestimation in the state value function, thereby providing more accurate allocations. Besides, this counterfactual advantage is always estimated using up-to-date joint policies, eliminating the necessity for the algorithm to account for changes in other agents' policies, thereby mitigating non-stationary issues in MARL.

Next, we present a theoretical analysis to illustrate that the counterfactual advantage can be used to optimize global returns. Without compromising the validity of our conclusions, we use $\boldsymbol{\pi}(\boldsymbol{a}|s) = \boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{x}(s)) = \Pi_{i=1}^{n} \pi_i(a_i|x_i)$ to denote the joint policy, as the mapping from s to $\boldsymbol{x}$ is injective. Hence, the expected return of the multi-agent system can be expressed as follows:

$$\eta(\boldsymbol{\pi}) = \mathbb{E}_{s_0, \boldsymbol{a}_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{10}$$

where $s_0 \sim p_0(s), \boldsymbol{a}_t \sim \boldsymbol{\pi}(\boldsymbol{a}_t|s_t), s_{t+1} \sim \mathbb{T}(s_{t+1}|s_t, \boldsymbol{a}_t)$, meaning that the data is sampled according to the policy $\boldsymbol{\pi}$.

Theorem 3.1. *For agent i in a multi-agent system, if its policy $\tilde{\pi}_i$ is obtained via improving the following objective, the optimization target in Eq. (10) can be improved:*

$$M'_{\boldsymbol{\pi}}(\tilde{\boldsymbol{\pi}}) = \eta(\boldsymbol{\pi}) + \sum_{s} p_{\boldsymbol{\pi}}(s) \sum_{\boldsymbol{a}} \tilde{\boldsymbol{\pi}}(\boldsymbol{a}|s) A_i^{\mathrm{MACD}} - C \cdot D_{\mathrm{KL}}^{\max}(\tilde{\pi}_i, \pi_i) \tag{11}$$

where $A_i^{\mathrm{MACD}} = Q_{\boldsymbol{\pi}}(s, \{a_i, \boldsymbol{a}_{-i}\}) - Q_{\boldsymbol{\pi}}(s, \{d, \boldsymbol{a}_{-i}\})$ is the counterfactual advantage, d is the default action, $\boldsymbol{\pi}$ is the current joint policy, $\tilde{\boldsymbol{\pi}}(\boldsymbol{a}|s) = \tilde{\boldsymbol{\pi}}(\boldsymbol{a}|\boldsymbol{x}(s)) = \tilde{\pi}_i(a_i|x_i)\boldsymbol{\pi}_{-i}(\boldsymbol{a}_{-i}|\boldsymbol{x}_{-i})$ is the updated policy.

The proof of this theorem can be found in Appendix A. This theorem indicates that agent policies can be improved by optimizing the counterfactual advantage. However, in practice, the loss function in Eq. (11) is hard to be implemented. Therefore, we employ the Proximal Policy Optimization (PPO) [24] algorithm to optimize agent policies in the policy learning phase. Furthermore, implementing sequential policy updates for agents entails substantial computational complexity, so we choose to update all agents' policies synchronously as an approximation of the theorem's conclusion. We train the actor by improving the following function:

$$\mathcal{L}_{\mathrm{actor}}^{\mathrm{MACD}}(\theta) = \mathbb{E}_t\Big[\sum_{i=1}^{n} \min(\rho_{i,t} A_{i,t}^{\mathrm{MACD}}, \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon) A_{i,t}^{\mathrm{MACD}})\Big] \tag{12}$$

where θ is the joint policy's parameter, ϵ is the clip coefficient, $\rho_{i,t} = \pi_i(a_{i,t}|\boldsymbol{x}_{i,t}; \theta)/\pi_i(a_{i,t}|\boldsymbol{x}_{i,t}; \theta_{\mathrm{old}})$ is the probability ratio that measures the difference between $\pi_i(\theta)$ and $\pi_i(\theta_{\mathrm{old}})$. In the vanilla PPO algorithm for multi-agent settings [37], the importance sampling ratio should be $\Pi_{i=1}^{n} \rho_{i,t}$ due to state values derived from old joint policies. However, our computation of counterfactual advantage values always uses up-to-date joint policies, mitigating non-stationarity by omitting the importance sampling term and thus stabilizing policy learning.

Algorithm 1: MACD Algorithm

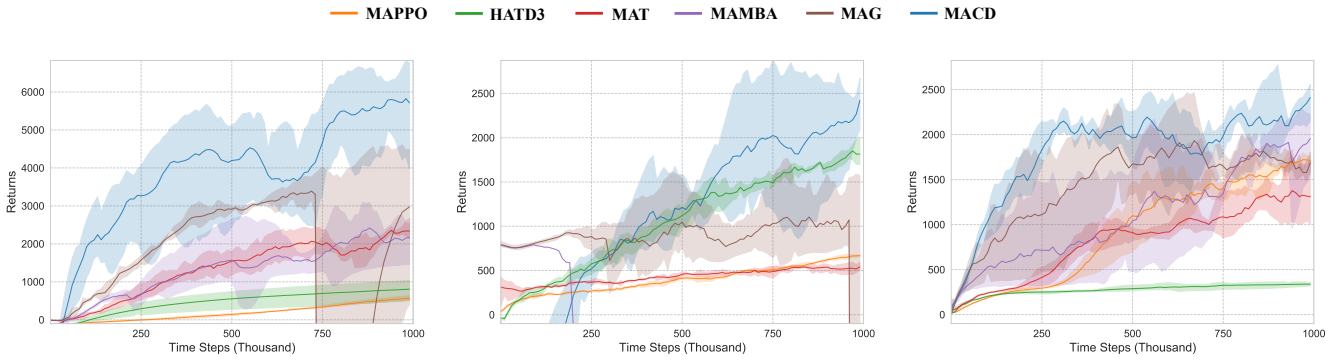
```
Initialize the parameters φ, θ, ψ, and a replay buffer M_env;
for episode ℓ = 1, 2, 3, ... do
    # Interacting with the True Environment
    Interact for an episode and store true data into M_env;
    # Model Learning Phase
    for model epoch = 1, 2, 3, ... do
        Sample transition sequence from M_env;
        Use world model to generate posterior latent states;
        Update φ by minimizing L(φ);
    end
    # Policy Learning Phase
    for imagination epoch = 1, 2, 3, ... do
        Sample transition sequence from M_env;
        Use world model to generate posterior latent states;
        Perform imagination rollout for pseudo interaction;
        for PPO epoch = 1, 2, 3, ... do
            Perform counterfactual rollout for A_{i,t}^MACD;
            Update θ by maximizing L_actor^MACD(θ) ;
            Update ψ by minimizing L_critic(ψ);
        end
    end
end
```

3.3 Overall Training Algorithm

The overall algorithm is built upon our CIDE framework, which combines the centralized imagination of the world model and the decentralized decision-making of the actor. As detailed in Algorithm 1, we conduct model and policy learning after the end of each episode. In the model learning phase, we sample some state transition sequences from $\mathcal{M}_{\mathrm{env}}$ to generate posterior latent states, which are subsequently utilized as inputs for the centralized predictors to jointly reconstruct global variables. In the policy learning phase, posterior latent states are also generated based on samples from $\mathcal{M}_{\mathrm{env}}$, serving as starting points for imagination rollouts. In the rollout process, the rewards and continuation flags are predicted in a centralized manner, and the actions are generated by the actors in a decentralized manner. This process generates a large amount of pseudo data used for policy training. It is worth noting that each policy update leverages counterfactual imaginations generated based on the up-to-date policies, aiming to promptly rectify the impact of changes in other agents' policies on agent i, thereby mitigating the non-stationary problem.

4 EXPERIMENTS

In this section, we conduct an empirical study of MACD on StarCraft Multi-Agent Challenge (SMAC) [23] and the multi-agent version of Mujoco (MA-Mujoco) [20]. Our evaluation involves comparisons between our approach and advanced model-free and model-based MARL algorithms. The empirical findings demonstrate the



(a) 2-Agent HalfCheetah (b) 4-Agent Ant (c) 3-Agent Hopper

Figure 3: Comparisons against SOTA model-free and model-based baselines on MA-Mujoco. Solid curves represent the mean of runs over 5 different random seeds, and shaded regions correspond to the confidence intervals. The evaluation metric is the average return of one episode, and MACD achieves the SOTA performance among all experimental methods.

Table 1: Mean and Std of the Winning Rates in SMAC

Map	Steps	MACD	MAG	MAMBA	MAPPO	QMIX	HASAC
2s3z	50k	**80**(15)	65(15)	60(9)	3(1)	24(4)	2(1)
3s_vs_4z	100k	**73**(16)	64(10)	34(8)	0(0)	0(0)	0(0)
MMM	100k	**83**(11)	25(5)	48(11)	2(1)	4(3)	0(0)
3s5z	200k	**60**(14)	15(4)	17(5)	17(5)	0(0)	0(0)
3s_vs_5z	200k	**57**(12)	17(11)	9(7)	0(0)	0(0)	0(0)
5m_vs_6m	200k	**48**(19)	47(12)	24(10)	0(0)	0(0)	0(0)

enhanced sample efficiency, training stability, and final cooperation performance achieved by our method. Additionally, we undertake a series of ablation experiments to substantiate the effectiveness of our CIDE framework and counterfactual advantage module.

4.1 Comparative Evaluation

Baselines. We conduct a comparative analysis of MACD with SOTA model-based MARL methods: MAMBA [6] and MAG [35]. MAMBA is notable as the first MARL algorithm that leverages the Dreamer framework, while MAG builds upon MAMBA's foundation by incorporating considerations of multi-step model rollout errors. We compare our approach with advanced model-free MARL methods, specifically MAPPO [37], HATD3 [41], and MAT [33] for continuous action spaces in MA-Mujoco experiments, and MAPPO, QMIX [22], and HASAC [41] for discrete action spaces in SMAC. The codes of these algorithms are all open-source, and the hyper-parameters are all optimized separately to attain their optimal performance.

Environment. We conduct experiments on SMAC and MA-Mujoco environments. SMAC is a cooperative multi-agent benchmark with discrete actions, involving two teams engaged in combat. One team is controlled by a game bot, while the other is managed by MARL algorithms. In contrast, MA-Mujoco is a widely-used cooperative multi-agent benchmark with continuous action space, where each scenario involves a robot with multiple joints organized into predefined groups and controlled by different agents. Agents are limited to observing the states of their assigned joints and can take actions to adjust their angles. Both environments share global rewards reflecting overall system performance. MA-Mujoco aims for higher robot speed with smaller action amplitudes, while SMAC pursues victory in combat. Details can be found in Appendix B.

Results. Table 1 displays SMAC results, indicating average winning rates and standard deviations (std) derived from five repeated experiments. MACD outperforms comparative methods within the given training timesteps. Figure 3 presents the learning curves of MA-Mujoco experiments, with each curve being the result of averaging five repeated experiments. HATD3 exhibits satisfactory performance on 4-agent Ant but fails in other scenarios. Conversely, model-based methods exhibit superior performance across a broader range of scenarios due to the presence of world models, overcoming limitations encountered by model-free methods related to exploration and other factors.

The shadows in the MAG and MAMBA's curves reveal their training instability due to the fully-decentralized world model, leading to insufficient global information for effective agent decision-making. Consequently, these algorithms sometimes fail to stabilize their training process, resulting in average returns even below -100000 in some trials. In contrast to all baseline methods, MACD demonstrates significantly superior sample efficiency and cooperation performance, while also maintaining the best stability throughout the experiments. Its utilization of the CIDE-based framework reduces true-world interactions and enhances training stability through the centralized imagination paradigm. The incorporation of the counterfactual module offers an enhanced approach to assign global rewards and address non-stationary challenges, ultimately leading to more efficient learning targets for the actors.

4.2 Ablation Studies

In this section, we conduct several ablation experiments to investigate the effect of our CIDE framework and the counterfactual module. Figure 4 shows the ablation experiment results on the 2-Agent HalfCheetah scenario of MA-Mujoco.

Counterfactual Advantage. We devise three ablation methods to underscore the significance of our counterfactual module. Specifically, MACD-RMCO entails the removal of the counterfactual module, aligning it with the conventional PPO advantage computation.

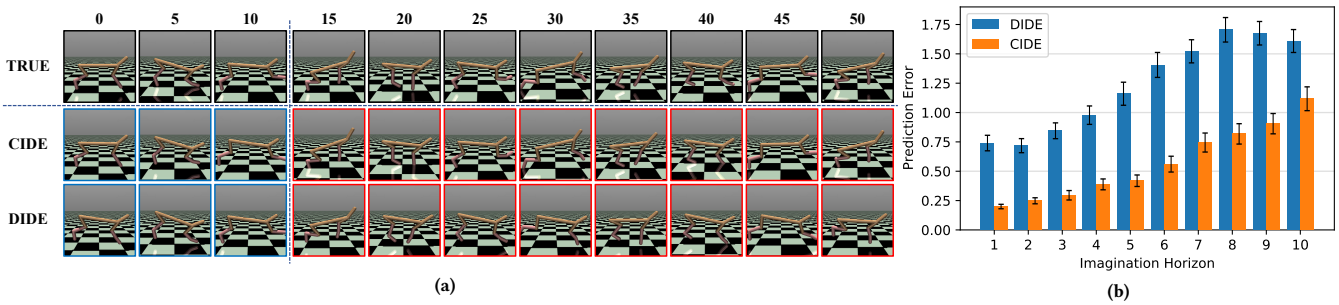


Figure 5: Prediction error analysis. (a) The long-term prediction demonstration on 2-Agent HalfCheetah. The first three images derive from real-world interaction data, while the following images stem from imagination rollout. All trajectories share identical action sequences and start from the same initial states and random seeds. (b) Imagination horizon's impact on prediction errors for robot global state with MACD-CIDE and MACD-DIDE methods.

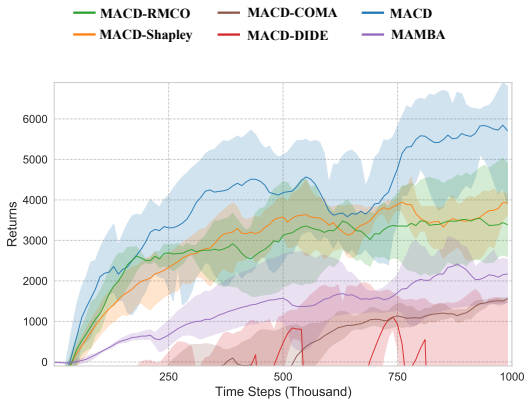


Figure 4: Ablation results of the CIDE framework and the counterfactual advantage module.

MACD-COMA and MACD-Shapley substitute our counterfactual module with other credit assignment methodologies, as detailed in Eq. (2) and Eq. (4), respectively. Experimental results show that the performance of MACD-Shapley is inferior to that of MACD, which could be attributed to the fact that designating the agent's action as the default action is not entirely equivalent to the agent exiting the game, thus failing to fulfill the conditions for the original Shapley theory, resulting in ineffective credit assignment. MACD-COMA's underperformance may be attributed to its inefficient marginal operation. Furthermore, MACD-RMCO exhibits inferior performance compared with MACD but surpasses MAMBA due to the presence of CIDE, which enhances the efficiency of the world model.

CIDE Framework. We remove the proposed CIDE framework and refer this method to MACD-DIDE (decentralized imagination with decentralized execution). Specifically, this method entails converting the global state predictor into local observation predictors. The world model of MACD-DIDE reconstructs local observations from each agent's perspective, which diminishes the representational capacity of latent states and subsequently diminishes the system's decision-making performance. This is evident in the curves presented in Figure 4. We also investigate the changes in global state prediction errors of CIDE and DIDE world models as the imagination horizon increases. Figure 5 (b) demonstrates that increasing the horizon results in a corresponding increase in prediction error, while CIDE exhibits significantly lower errors compared with DIDE. The mean and error bars are calculated from 256 random episodes.

To further demonstrate the superiority of our CIDE framework, we utilize Figure 5 to visually depict the state sequence in the 2-Agent HalfCheetah scenario. The first line in Figure 5 represents ground-truth robot states, and the second and third lines represent robot states reconstructed by CIDE-based and DIDE-based MACD, respectively. We divide a single episode into two distinct stages. The first stage spans timesteps 0 to 14, during which the agents interact with the true environment. During this stage, the agents can utilize observations to generate posterior latent state z_t using the representation model for robot state reconstruction, as indicated by blue boxes in Figure 5. As depicted, both the CIDE and DIDE frameworks effectively predict the global state. The second stage encompasses timestep 15 to 50, where the agents utilize the transition predictor of the learned world model to conduct imagination rollouts. During this stage, the agents employ imagined latent states $\hat{z}_t$ for robot state prediction, as red boxes indicate. The trajectories derived from both the CIDE and DIDE method use identical action sequences, acquired from true environmental decisions. Figure 5 indicates that the CIDE-based method excels in achieving higher performance for long-term predictions, while the DIDE-based framework maintains accuracy only up to ten timesteps. These findings underscore the superior global state reconstruction and prediction capabilities of our CIDE framework, supporting pseudo data of higher quality.

Counterfactual Horizon. To further investigate the impact of our counterfactual module, we conduct an ablation experiment by varying the counterfactual rollout horizon H_c. A horizon of zero entails estimating the counterfactual advantage solely through the critic network, akin to COMA, whose performance is greatly influenced by its critic networks. Increasing the horizon appropriately enhances the accuracy of value estimations, as reward predictions are obtained through supervised learning, thereby providing greater precision. Figure 6 illustrates that as the horizon increases from 0 to 8, the algorithm's average return improves. However, further increasing the horizon leads to a decline in performance due to accumulating errors in the world model, which affects the accuracy of

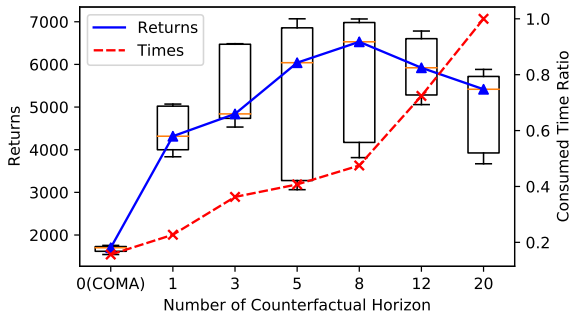


Figure 6: Counterfactual horizon's impact on the average return and the training time.

counterfactual advantage estimation. We normalize the consumed hours relative to the maximum value in the red curve of Figure 6. The results indicate that when the horizon is set to 20, the required training time for the same timesteps is five times longer than when the horizon is 1. Considering both performance and time complexity, the algorithm's optimal performance occurs at a horizon of 8. However, compared with other algorithms, superior performance is still achieved with a horizon greater than 1.

5 RELATED WORKS

Inspired by the progress made by model-based RL methods in single-agent settings, recent works attempt to apply model-based RL techniques to MARL training. AORPO [40] employs a decentralized environment model and opponent models to improve sample efficiency via adaptive opponent-wise rollouts. MAMBPO [34] formulates a model-based MARL approach, exhibiting superior sample efficiency in multi-robot tasks. MBOM [38] considers the improvement of the opponent policies and uses the environment model to obtain the best response. Besides, CPS [2] employs a coordination graph to identify the state-action pairs in the most urgent need of updating, effectively addressing the cooperation challenges in large-scale and discrete environments. MDBS [36] introduces model-based shields to monitor and rectify unsafe behaviors in MARL agents. These works adopt a naive framework for their world models, leading to the low efficiency of the rollout in the imagination space. MAMBA extends Dreamer from the single-agent setting to the multi-agent setting [6]. Leveraging Dreamer's long-term prediction capabilities in policy training, MAMBA achieves superior sample efficiency compared to model-free MARL methods. MAG [35] extends MAMBA by treating model rollout as a multi-step decision-making process, mitigating the cumulative errors in pseudo data. However, their fully-decentralized MA-RSSM leads to imprecise prediction of the global state transition of the whole system. In contrast, MACD incorporates the CIDE framework to tackle this issue, resulting in an improvement in pseudo data quality. Moreover, these methods have not fully leveraged the potential of world models in addressing fundamental challenges in MARL training, including the non-stationary and credit assignment issues.

Some MARL methods try to address the credit assignment problem by allocating the contribution for each agent. QMIX [22] and its modifications [8, 21, 26] adopt a joint value function to guide the update of agent policies for implicit credit assignment. Explicit methods try to use the impact of agent actions on global rewards to evaluate the contribution of agents to the whole system. COMA [7] is a model-free MARL method that adopts the joint value function to marginalize agent actions, thus evaluating its contribution via the counterfactual advantage. SQDDPG [32] uses the Shapley Q value as the critic in the deep deterministic policy gradient algorithm to distribute the global reward. SHAQ [31] derives the Shapley-Bellman operation to further fill the gap in SQDDPG on theoretical guarantees of convergence. DAE [17] establishes a model-based reward predictor and reshapes the global reward into a potential-based difference reward. The theoretical and empirical results show its performance in credit assignment. *Han et. al* [12] present a one-step world model to compute the Shapley advantage for each agent to guide the update of decentralized policies. However, these methods rely on state value functions, which inaccurately estimate values beyond the current policy distribution, leading to imprecise contribution allocation. In contrast, MACD employs a supervise-learned world model to establish a model-based counterfactual module for more efficient and precise contribution evaluation.

6 CONCLUSION

This study introduces MACD, a novel model-based MARL approach addressing sample efficiency, credit assignment, and non-stationary challenges via world models. We present a centralized imagination with decentralized execution (CIDE) framework to extend the multi-agent version of RSSM. The centralized predictors within this framework contribute to the reconstruction of the overall system's state transition process, thereby improving the generation of higher-quality pseudo data and, consequently, enhancing sample efficiency. Furthermore, to address the credit assignment and non-stationary challenges, MACD employs a counterfactual module to evaluate an agent's contribution to the whole system. This module replaces an agent's actions with default actions and calculates their impact on global rewards through an additional counterfactual trajectory. A theoretical analysis is established to prove that this counterfactual policy update rule maximizes the multi-agent learning objective. Comparative evaluations between MACD and SOTA model-free and model-based methods demonstrate MACD's ability to consistently achieve optimal sample efficiency and cooperation performance across multiple scenarios. MACD also obtains the most stable training performance compared with MAMBA and MAG on MA-Mujoco experiments. We validate the efficacy of our proposed CIDE framework and counterfactual module through exhaustive ablation studies and visualization demonstration. Furthermore, we analyze the influence of the counterfactual horizon on the method's performance. In future work, we intend to harness MACD's precise prediction capabilities to address real-world challenges.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under Grant 62136008 and 62293541, Beijing Natural Science Foundation under Grant No. 4232056, Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27030400, Youth Innovation Promotion Association CAS (2021132), and IEEE CIS Graduate Student Research Grant.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Eugenio Bargiacchi, Timothy Verstraeten, and Diederik M. Roijers. 2021. Cooperative prioritized sweeping. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '21)*. 160–168.

[3] Jiajun Chai, Weifan Li, Yuanheng Zhu, Dongbin Zhao, Zhe Ma, Kewu Sun, and Jishiyu Ding. 2023. UNMAS: Multiagent reinforcement learning for unshaped cooperative scenarios. *IEEE Transactions on Neural Networks and Learning Systems* 34, 4 (2023), 2093–2104.

[4] Jiajun Chai, Yuanheng Zhu, and Dongbin Zhao. 2023. NVIF: Neighboring variational information flow for cooperative large-scale multiagent reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems* (2023), 1–13.

[5] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[6] Vladimir Egorov and Alexei Shpilman. 2022. Scalable multi-agent model-based reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22)*. 381–390.

[7] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[8] Matteo Gallici, Mario Martin, and Ivan Masmitja. 2023. TransfQMix: Transformers for leveraging the graph structure of multi-agent reinforcement learning problems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. 1679–1687.

[9] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2020. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*.

[10] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2021. Mastering Atari with discrete world models. In *International Conference on Learning Representations*.

[11] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2023. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104* (2023).

[12] Dongge Han, Chris Xiaoxuan Lu, Tomasz Michalak, and Michael Wooldridge. 2022. Multiagent model-based credit assignment for continuous control. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22)*. 571–579.

[13] Guangzheng Hu, Yuanheng Zhu, Dongbin Zhao, Mengchen Zhao, and Jianye Hao. 2023. Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems* 34, 8 (2023), 3966–3978.

[14] Ehsan Imani and Martha White. 2018. Improving regression performance with distributional losses. In *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80. 2157–2166.

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

[16] Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. 2021. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 934–942.

[17] Yueheng Li, Guangming Xie, and Zongqing Lu. 2022. Difference advantage estimation for multi-agent policy gradients. In *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162. 13066–13085.

[18] Jinming Ma and Feng Wu. 2023. Learning to coordinate from offline datasets with uncoordinated behavior policies. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 1258–1266.

[19] Hangyu Mao, Wulong Liu, Jianye Hao, Jun Luo, Dong Li, Zhengchao Zhang, Jun Wang, and Zhen Xiao. 2020. Neighborhood cognition consistent multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7219–7226.

[20] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. 2021. FACMAC: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems* 34 (2021), 12208–12221.

[21] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. 2020. Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems* 33 (2020), 10199–10210.

[22] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic value function factorisation for deep multi-agent Rrinforcement learning. In *International Conference on Machine Learning*. 4295–4304.

[23] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19)*. 2186–2188.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[25] Lloyd S Shapley. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39, 10 (1953), 1095–1100.

[26] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. 5887–5896.

[27] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18)*. 2085–2087.

[28] Zhentao Tang, Yuanheng Zhu, Dongbin Zhao, and Simon M. Lucas. 2023. Enhanced rolling horizon evolution algorithm with opponent model learning: Results for the fighting game AI competition. *IEEE Transactions on Games* 15, 1 (2023), 5–15.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30. 5998–6008.

[30] Junjie Wang, Qichao Zhang, and Dongbin Zhao. 2022. Dynamic-horizon model-based value estimation with latent imagination. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–14.

[31] Jianhong Wang, Yuan Zhang, Yunjie Gu, and Tae-Kyun Kim. 2022. SHAQ: Incorporating Shapley value theory into multi-agent Q-learning. *Advances in Neural Information Processing Systems* 35 (2022), 5941–5954.

[32] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Shapley Q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7285–7292.

[33] Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems* 35 (2022), 16509–16521.

[34] Daniël Willemsen, Mario Coppola, and Guido CHE de Croon. 2021. MAMBPO: Sample-efficient multi-robot reinforcement learning using learned world models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5635–5640.

[35] Zifan Wu, Chao Yu, Chen Chen, Jianye Hao, and Hankz Hankui Zhuo. 2023. Models as agents: Optimizing multi-step predictions of interactive local models in model-based multi-agent reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 9 (2023), 10435–10443.

[36] Wenli Xiao, Yiwei Lyu, and John Dolan. 2023. Model-based dynamic shielding for safe and efficient multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. 1587–1596.

[37] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*. 24611–24624.

[38] Xiaopeng Yu, Jiechuan Jiang, Wanpeng Zhang, Haobin Jiang, and Zongqing Lu. 2022. Model-based opponent modeling. *Advances in Neural Information Processing Systems* 35 (2022), 28208–28221.

[39] Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, and Junliang Xing. 2023. Sequential cooperative multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. 485–493.

[40] Weinan Zhang, Xihuai Wang, Jian Shen, and Ming Zhou. 2021. Model-based multi-agent policy optimization with adaptive opponent-wise rollouts. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence* (2021).

[41] Yifan Zhong, Jakub Grudzien Kuba, Siyi Hu, Jiaming Ji, and Yaodong Yang. 2023. Heterogeneous-agent reinforcement learning. *arXiv preprint arXiv:2304.09870* (2023).

[42] Hanhan Zhou, Tian Lan, and Vaneet Aggarwal. 2022. PAC: Assisted value factorization with counterfactual predictions in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 15757–15769.

[43] Yuanheng Zhu, Weifan Li, Mengchen Zhao, Jianye Hao, and Dongbin Zhao. 2023. Empirical policy optimization for n-player markov games. *IEEE Transactions on Cybernetics* 53, 10 (2023), 6443–6455.

[44] Yuanheng Zhu and Dongbin Zhao. 2022. Online minimax Q network learning for two-player zero-sum Markov games. *IEEE Transactions on Neural Networks and Learning Systems* 33, 3 (2022), 1228–1241.