Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CTMAPPO-Clip: A CTCE-Based Approach to Mitigate Policy Overfitting in Multi-Agent Reinforcement Learning

Qile Bo
Jiangsu University of Science and Technology
Dongsheng Wang
jsjxy_wds@just.edu.cn
Jiangsu University of Science and Technology
Xinghao Han
JiangSu Automation Research Institute
Liang Shi
Jiangsu University of Science and Technology
Yue Niu
Jiangsu University of Science and Technology
Huige Li
Jiangsu University of Science and Technology
Tianning Zhang
Jiangsu University of Science and Technology
Yun Cui
Jiangsu University of Science and Technology

Research Article

Keywords: Reinforcement Learning, Multi-Agent, PPO, CTCE paradigm

Posted Date: April 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9241599/v1

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

Additional Declarations: No competing interests reported.

CTMAPPO-Clip: A CTCE-Based Approach to Mitigate Policy Overfitting in Multi-Agent Reinforcement Learning

Qile Bo[1], Dongsheng Wang[1], Xinghao Han[2], Liang Shi[1], Yue Niu[1], Huige Li[1], Tianning Zhang[1], Yun Cui[1], Shu Pan[1]

[1]Jiangsu University of Science and Technology, Zhenjiang 212000, China.
[2]JiangSu Automation Research Institute, Lianyungang 222000, China.

Abstract

Achieving efficient coordination among multiple agents has become a key research focus in reinforcement learning. However, in complex collaborative environments such as SMAC, existing approaches often suffer from limited policy generalization and a tendency toward policy overfitting during multi-agent coordination. To address these limitations, we propose CTMAPPO-Clip, a novel algorithm based on the CTCE paradigm, designed to mitigate policy overfitting in MARL. First, we model and optimize the joint policy by decomposing it into a set of conditional probability distributions. This allows each agent to independently optimize its own policy function given the observed states and actions of other agents, thereby improving policy generalization and decision-making stability. Second, we incorporate a Transformer-based architecture into the policy network, leveraging self-attention mechanisms to capture inter-agent dependencies and collaborative patterns, thereby enhancing the expressive power and coordination modeling of the joint policy. Finally, we introduce an advantage clipping mechanism, which dynamically truncates excessively high advantage values during training. This suppresses noisy gradient updates caused by outlier advantages, reduces the risk of converging to suboptimal policies, and enhances the robustness of policy learning. Experimental results on the SMAC benchmark demonstrate that CTMAPPO-Clip outperforms several state-of-the-art baselines, including QMIX and MAPPO, achieving superior performance and validating the effectiveness of CTMAPPO-Clip in mitigating policy overfitting in MARL.

Keywords: Reinforcement Learning, Multi-Agent, PPO, CTCE paradigm

1 Introduction

In recent years, Multi-Agent Reinforcement Learning (MARL) has made significant progress in complex collaborative tasks [1–4]. While decentralized execution strategies [5] and the Centralized Training with Decentralized Execution (CTDE) paradigm [6] have improved learning stability, they limit the use of global information during execution. In contrast, the Centralized Training with Centralized Execution (CTCE) paradigm [7] utilizes global information during both training and execution [8, 9], effectively mitigating instability caused by partial observability [10]. However, CTCE faces a major scalability bottleneck: the dimensionality of the joint action space increases exponentially with the number of agents [11]. Furthermore, representative algorithms like MAPPO [12], despite their success in benchmarks like SMAC [13], still suffer from policy overfitting in these high-dimensional joint action spaces [14]. This overfitting typically stems from advantage-based updates misguiding agents toward suboptimal policies, a challenge that manual feature engineering methods [15] struggle to address generally.

We propose CTMAPPO-Clip, a CTCE-based algorithm designed to mitigate policy overfitting in MAPPO. The main contributions of this work are as follows:

(1) We propose a joint policy modeling method based on conditional probability decomposition. Instead of directly optimizing a high-dimensional joint policy, CTMAPPO-Clip factorizes it into an autoregressive sequence of conditional policies, allowing each agent to generate actions conditioned on global observations and previously generated teammate actions. This design reduces optimization difficulty and improves policy generalization and decision stability.

(2) We integrate a Transformer architecture into the policy network to strengthen cooperative representation learning. The self-attention module extracts global interaction features among agents, while the cross-attention module models action dependencies during sequential action generation, improving the expressive power of the joint policy in complex coordination tasks.

(3) We introduce a dynamic advantage clipping mechanism for robust policy optimization. By constructing an adaptive clipping boundary from batch-level advantage statistics, the method suppresses noisy updates caused by outlier advantages, stabilizes training, and reduces the risk of convergence to suboptimal local policies.

2 Related Works

In recent years, MARL has made significant progress in cooperative tasks and has been widely applied in areas such as robotic control, distributed systems, and game-theoretic environments. To address the challenges of non-stationarity and credit assignment among agents [16], researchers have proposed various training paradigms and algorithmic frameworks.

2.1 Representative Algorithms Based on Value Decomposition Methods

Among value decomposition methods, representative algorithms such as QMIX [17] and Value Decomposition Networks (VDN) [18] have shown strong performance on the SMAC benchmark environment. These approaches adopt the CTDE paradigm and decompose the global action-value function into individual agent-wise value functions to enable effective coordination. However, due to structural constraints, they have limited capacity to model complex inter-agent dependencies and suffer from significant credit assignment bias [19]. To improve the expressive capacity and quality of policy optimization, policy gradient methods have gradually become mainstream. In particular, PPO achieved great success in single-agent tasks, leading to widespread interest in its multi-agent extensions. Independent PPO (IPPO) [20] applies single-agent PPO independently to each agent to perform decentralized policy optimization, but it fails to address the instability caused by inter-agent non-stationarity[21]. Multi-Agent PPO (MAPPO) integrates centralized value functions within the CTDE framework to guide policy learning, resulting in better coordination. However, studies have shown [22] that MAPPO suffers from training instability and policy overfitting in SMAC scenarios, which limits its generalization ability and reliability.

2.2 Single-Agent Approaches for Mitigating Overfitting

To mitigate policy overfitting, several improvements have been proposed. Recent studies[23] show that fine-tuned QMIX can alleviate overfitting to some extent. As a value decomposition algorithm, QMIX approximates the global Q-function by combining individual Q-values through a monotonic mixing function while preserving optimality, enabling efficient collaboration. However, in practice, QMIX is susceptible to overestimation bias [24], which can negatively affect stability and convergence. To address this issue, researchers proposed Regularized Softmax Deep Multi-Agent Q-Learning [25], which introduces regularization terms and softmax operations to mitigate overestimation and achieves state-of-the-art performance across multiple cooperative benchmarks. Despite the success of CTDE in MARL applications, its decentralized execution limits inter-agent information sharing. In fully cooperative environments where global information is available, this limitation often leads to insufficient information utilization and inconsistent policy convergence. To address this, the CTCE paradigm has been proposed, allowing agents to access global observations and make joint decisions during both training and execution phases, thereby enabling more stable and efficient cooperative control. However, CTCE also faces major challenges, most notably the exponential growth of the joint action space with the number of agents. This results in considerable computational and memory overhead in large-scale systems, restricting the practicality of CTCE in real-world applications.

2.3 Multi-Agent Approaches for Mitigating Overfitting

To further alleviate policy overfitting in MARL, researchers have explored various regularization techniques and architectural enhancements. For instance, Xu et al. [26] proposed a KL-divergence-based policy regularization approach to limit the extent of

policy updates, effectively reducing overfitting. Bai et al. [27] introduced population-based training, which simulates the coexistence of multiple policy variants to improve the robustness of the learning process. While these methods enhance generalization to some extent, they often come at the cost of significantly increased model complexity, which compromises deployment efficiency. More recently, the Transformer architecture has demonstrated promising results in MARL and has become a powerful tool for modeling inter-agent dependencies. For example, Feng et al. [28] proposed a Transformer-based multi-agent communication framework that employs self-attention to model communication structures among agents. Hu et al. [29] further introduced Transformer-based modeling for joint policy learning, enhancing both expressive power and generalization performance. These efforts provide new directions for addressing the challenges of high-dimensional joint policy modeling and the curse of dimensionality in joint action spaces. While CTDE remains more scalable in large systems, CTCE offers greater potential in modeling collaborative behavior. Effectively mitigating policy overfitting during optimization–while also balancing model performance and computational efficiency–remains one of the central challenges in multi-agent reinforcement learning research.

Relation to Multi-Agent Transformer (MAT): The paradigm of treating MARL as a sequence modeling problem has also been explored in MAT. Similar to our policy network, MAT employs an encoder-decoder architecture to factorize the joint policy into a sequence of conditional probabilities. However, CTMAPPO-Clip differs from MAT in two key aspects:

Objective and Optimization Mechanism: While MAT focuses on leveraging the Transformer to find optimal joint policies, CTMAPPO-Clip is specifically designed to mitigate policy overfitting in high-dimensional action spaces. To this end, we introduce a novel dynamic advantage clipping mechanism in the policy optimization objective. Unlike standard PPO updates used in MAT, our approach dynamically truncates outlier advantage values, preventing the policy from converging to suboptimal local minima driven by noisy signals.

Critic Architecture for Temporal Modeling: Instead of using a Transformer-based critic, CTMAPPO-Clip employs a specialized MLP-GRU-MLP architecture for value estimation. This hybrid design enables more efficient capture of long-term temporal dependencies and environmental non-stationarity through recurrent units, which complements the spatial attention mechanism of the actor.

3 Methods

The overall architecture of the proposed CTMAPPO-Clip, a CTCE-based multi-agent reinforcement learning algorithm, is illustrated in Figure 1. CTMAPPO-Clip consists of three main components: a policy network for joint policy modeling, a value-estimation critic network, and an advantage clipping mechanism.

First, for joint policy modeling, we design a parameter-shared policy network that factorizes the joint policy into a set of conditional probability distributions. This formulation enables each agent to independently optimize its own policy function conditioned on the observed states and actions of other agents, thereby improving

generalization capability and the stability of cooperative decision-making. The policy network is built upon a Transformer-based architecture, which leverages self-attention mechanisms to effectively capture inter-agent dependencies and coordination patterns, thus enhancing both the expressive power and the coordination modeling capacity of the joint policy. Second, for value estimation, CTMAPPO-Clip employs a critic network to model the value function over joint actions. This critic network takes as input the global state information and joint action embeddings, and incorporates an MLP-GRU-MLP architecture [30] to effectively capture temporal features and model complex dependencies between states and actions. The output is a global value estimate corresponding to the current state, which serves as a critical learning signal for policy updates. Finally, an advantage clipping mechanism is introduced to dynamically truncate anomalously large advantage values during the computation of the advantage function. This limits the excessive influence of outlier advantages on policy updates, effectively reducing the risk of converging to local optima and improving both the robustness and convergence stability of policy learning.

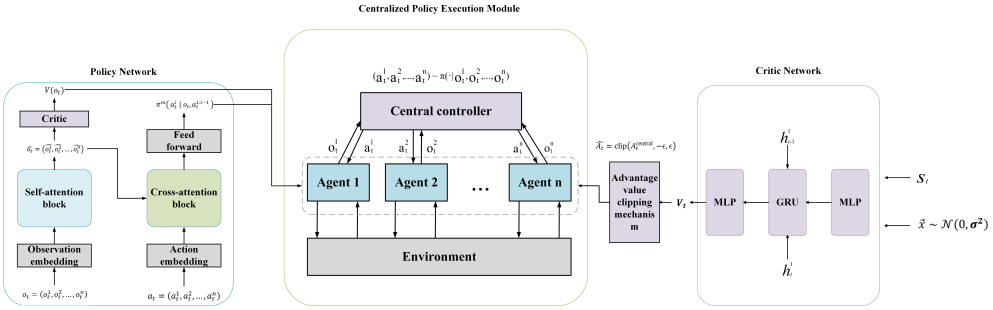


Fig. 1 The overall architecture of the CTMAPPO-Clip

3.1 Conditional-Probability-Decomposed Joint Policy Modeling with Transformer

In cooperative MARL, directly modeling the full joint action distribution quickly becomes intractable as the number of agents increases. CTMAPPO-Clip therefore models joint decision-making as an autoregressive conditional generation process under CTCE. Let $o_t = (o_t^1, \ldots, o_t^n)$ denote the joint observation and $a_t = (a_t^1, \ldots, a_t^n)$ denote the joint action at time step t. The joint policy is factorized by the chain rule as:

$$\pi_\theta(a_t \mid o_t) = \prod_{i=1}^{n} \pi_\theta(a_t^i \mid o_t, a_t^{1:i-1}) \tag{1}$$

where $a_t^{1:i-1}$ is the set of previously generated actions in the decoding order.

This decomposition provides two benefits: (i) it converts a high-dimensional joint optimization problem into a sequence of conditional prediction problems, and (ii) it introduces explicit causal dependency through $a_t^{1:i-1}$, so later decisions are conditioned

on earlier teammate decisions. We use a parameter-sharing actor across agents, so all conditional policies are generated by the same parameter set θ, improving scalability and consistency.

To realize this decomposition, CTMAPPO-Clip adopts a Transformer encoder–decoder architecture.

(1) Observation encoding with self-attention. Each local observation o_t^i is mapped by a shared embedding network f_{emb} to e_t^i, and the stacked embedding matrix is $E_t = [e_t^1, \ldots, e_t^n]$. The encoder computes:

$$Q = E_t W^Q, \quad K = E_t W^K, \quad V = E_t W^V \tag{2}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{3}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$. After multi-head attention and feed-forward blocks, we obtain the global interaction-aware representation $\hat{O}_t$. This allows each agent to focus on relevant teammates (e.g., support units attending to low-health allies). The representation $\hat{O}_t$ is used both as the contextual input to the autoregressive decoder and as a global representation for centralized value estimation.

(2) Autoregressive action decoding with cross-attention. At decoding step i, previously generated actions are causally masked before being embedded as the query sequence, ensuring order consistency and preventing information leakage. The context feature is computed by:

$$c_t^i = \text{CrossAttention}\Big(\text{Query} = E(a_t^{1:i-1}), \text{Key} = \hat{O}_t, \text{Value} = \hat{O}_t\Big) \tag{4}$$

Then the conditional policy for agent i is:

$$\pi_\theta(a_t^i \mid o_t, a_t^{1:i-1}) = \text{Softmax}\big(\text{MLP}(c_t^i)\big) \tag{5}$$

Thus, each action is generated using both global context and prior teammate decisions, which improves coordination in high-dimensional joint action spaces.

(3) Connection to centralized value-guided training. Under the same CTCE formulation, centralized advantage signals are computed for policy updates:

$$A_t^{\text{central}} = r_t + \gamma V_\phi(s_{t+1}, a_{t+1}) - V_\phi(s_t, a_t) \tag{6}$$

This shared global baseline further stabilizes optimization; the detailed critic architecture is described in the next subsection.

3.2 Centralized Value Estimation via an MLP-GRU-MLP Critic

To provide reliable global training signals for policy updates, CTMAPPO-Clip adopts a centralized critic with an MLP-GRU-MLP architecture. During training, the critic takes the global state s_t and joint action a_t as input and outputs a scalar value estimate.

Table 1 Network Architecture and Dimension Flow of the Critic Network

Module	Layer Type	Input Dim.	Output Dim.	Activation	Description
State-Action Encoding	Concatenation	-	$d_s + d_a^{joint}$	-	Concat state & actions
	FC Layer 1	$d_s + d_a^{joint}$	d_{hidden}	ReLU	Feature extraction
	FC Layer 2	d_{hidden}	d_{hidden}	ReLU	Feature refinement
Temporal Modeling	GRU Cell	d_{hidden}	d_{hidden}	Tanh	Recurrent update
Global Value Prediction	FC Layer 3	d_{hidden}	d_{hidden}	ReLU	Post-processing
	Output Layer	d_{hidden}	1	Linear	Final value estimate

The critic first builds a joint feature representation:

$$x_t = \phi(s_t, a_t) \tag{7}$$

where $\phi(\cdot)$ is an MLP-based embedding function.

Temporal dependencies are modeled by:

$$h_t = \mathrm{GRU}(\mathrm{MLP}_1(x_t), h_{t-1}) \tag{8}$$

The global value estimate is then produced by:

$$V_\phi(s_t, a_t) = \mathrm{MLP}_2(h_t) \tag{9}$$

Using temporal-difference estimation, the centralized advantage is:

$$A_t^{\mathrm{central}} = r_t + \gamma V_\phi(s_{t+1}, a_{t+1}) - V_\phi(s_t, a_t) \tag{10}$$

This shared critic signal reduces variance and improves optimization stability for multi-agent policy learning.

3.3 Dynamic Advantage Clipping Mechanism

To further reduce policy overfitting in complex cooperative tasks, CTMAPPO-Clip introduces a dynamic advantage clipping mechanism. In multi-agent environments, centralized advantages can exhibit large variance due to stochastic transitions, non-stationary teammate policies, and exploratory behavior. Because the advantage distribution changes across training stages and scenarios, a fixed absolute threshold is often either too loose (ineffective clipping) or too strict (over-clipping useful gradients).

We therefore construct an adaptive clipping boundary from batch statistics. Let σ_A denote the standard deviation of A_t^{central} in the current mini-batch and let ξ be a base clipping coefficient. The clipped advantage is defined as:

$$\hat{A}_t = \mathrm{clip}(A_t^{\mathrm{central}}, -\xi\sigma_A, \xi\sigma_A) \tag{11}$$

This operation adaptively constrains advantages in a variance-aware interval, filtering out extreme values caused by estimation noise and reducing the risk of unstable updates and convergence to local optima.

Using $\hat{A}_t$, we optimize the clipped PPO objective:

$$L^{\text{clip}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{12}$$

where the policy ratio is

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid o_t)}{\pi_{\theta_{\text{old}}}(a_t \mid o_t)} \tag{13}$$

The $\min(\cdot)$ operator enforces conservative updates: once the policy change exceeds the clipping interval, additional positive incentives are suppressed.

The value regression term is

$$L^{\text{value}}(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_t, a_t) - R_t \right)^2 \right] \tag{14}$$

where R_t is the return target (e.g., TD/bootstrapped return).

The final loss is

$$L = L^{\text{clip}}(\theta) + c_1 L^{\text{value}}(\phi) - c_2 \mathbb{E}_t \left[\mathcal{H}(\pi_\theta(\cdot \mid o_t)) \right] \tag{15}$$

For discrete action spaces,

$$\mathcal{H}(\pi_\theta(\cdot \mid o_t)) = -\sum_{a \in \mathcal{A}} \pi_\theta(a \mid o_t) \log \pi_\theta(a \mid o_t) \tag{16}$$

where c_1 and c_2 balance value learning and entropy regularization. Overall, this mechanism stabilizes gradient updates and improves robustness and convergence in challenging multi-agent scenarios.

3.4 Computational Complexity Analysis

To evaluate the efficiency and scalability of CTMAPPO-Clip under the CTCE paradigm, we provide a complexity analysis of three major components: policy forward computation, centralized critic computation, and backward parameter updates.

(1) Forward-pass complexity of the policy network. Let n be the number of agents, T be the number of time steps per episode, d_o be the observation dimension, d_h be the hidden dimension, and d_a be the action dimension. The upper-bound complexity of policy forward computation is:

$$\mathcal{O}\big(n\,T\,\big(d_o d_h + d_h^2 + d_h d_a\big)\big) \tag{17}$$

(2) Complexity of the centralized value function. Let d_s denote the global-state dimension and d_a^{joint} denote the joint-action dimension. For the MLP-GRU-MLP

critic, the forward complexity is:

$$\mathcal{O}\big(T\left(d_s d_h + d_a^{\text{joint}} d_h + d_h^2\right)\big) \tag{18}$$

Since the centralized critic is only used during training, this overhead does not affect real-time execution.

(3) Back-propagation complexity of policy and critic updates. Let $|\theta|$ and $|\phi|$ be the parameter sizes of the policy and critic networks, respectively. The complexity of one backward-update pass is approximated by:

$$\mathcal{O}((|\theta| + |\phi|)T) \tag{19}$$

Overall, CTMAPPO-Clip maintains manageable training-time complexity while improving robustness against policy overfitting.

4 Experiment

4.1 Experimental Environment

SMAC is a widely used benchmark environment for evaluating advanced multi-agent learning methods. It provides a series of StarCraft II unit micromanagement scenarios with varying levels of difficulty. In these scenarios, a team of algorithm-controlled units must defeat an enemy team controlled by built-in AI. The units in SMAC are diverse, with homogeneous scenarios involving teams of units of the same type, and heterogeneous scenarios involving teams of units of different types. SMAC offers various scenarios categorized into easy, hard, and super hard difficulty levels, with each difficulty containing both homogeneous and heterogeneous scenarios. For our experiments, we use the Hard and Super Hard scenarios in SMAC as the primary benchmark settings.

4.2 Comparison Algorithms

In this section, we select two representative comparison algorithms in the current field of multi-agent reinforcement learning: the fine-tuned QMIX and MAPPO. QMIX is a value decomposition-based CTDE method. The core idea is to introduce a mixing network that decouples the individual Q-values of each agent in an additive manner, thereby approximating the joint action-value function. This approach allows each agent to make decisions based on local observations without sharing policies. The mixing network relies on global state information during training to ensure optimality constraints when combining individual Q-values into a joint Q-value. MAPPO is a policy-gradient-based method that adopts the CTDE framework. At each time step, the algorithm inputs all agents' observations and the global state into a shared Critic to perform policy optimization. The Actor is typically modeled independently for each agent to generate specific action policies. By using these two methods, we can systematically compare the performance of multi-agent reinforcement learning tasks from different algorithmic paradigms.

4.3 Hyperparameter Settings for CTMAPPO-Clip

In terms of training steps, the total number of training steps for the SMAC benchmark experiment is set to 10 million steps. During the training process, test evaluations are performed at regular intervals to plot the median test win rate curve, which helps track the strategy's convergence process and performance trends. For the proposed dynamic advantage clipping mechanism, we define a base tolerance factor ξ to construct the dynamic truncation boundary $[-\xi \cdot \sigma_A, \xi \cdot \sigma_A]$. If ξ is too large, the clipping mechanism fails to suppress extreme gradients; if ξ is too small, it restricts the propagation of valid gradient information and slows policy learning. Through extensive preliminary experiments, we set the base tolerance factor to $\xi = 3.0$. This adaptive boundary strikes a good balance between suppressing abnormal gradient fluctuations and maintaining sufficient gradient flow, ensuring both training stability and strong final performance.

4.4 Hardware Environment

To strike a balance between high performance and reproducibility, all algorithms in this study were trained and evaluated on a standardized hardware platform. The hardware resources required for the experiments include high-performance GPUs, multi-core CPUs, and sufficient memory to accommodate the high-concurrency computations and memory usage typical of multi-agent training. The detailed hardware configuration of the experimental platform is presented as follows.

Table 2 Hardware configuration of the experimental platform

Hardware Type	Specifications
CPU	Intel Xeon Gold 6226R @ 2.90GHz
GPU	NVIDIA RTX 3090 24GB
RAM	128GB
Deep Learning Frameworks	PyTorch 2.0.1
Multi-agent environment	SMAC v1.0

All the code for the experiments was run on the same platform to ensure that the comparison algorithms were trained under consistent conditions, thus ensuring the fairness and reliability of the experimental results.

4.5 Evaluation Metrics

To comprehensively assess the performance of the proposed CTMAPPO-Clip in multi-agent cooperative scenarios, the average test win rate is used as the primary evaluation metric. This metric provides an intuitive measure of the agents' ability to complete tasks and their cooperation effectiveness during the testing phase. It has been widely adopted in the evaluation of multi-agent benchmark environments such as SMAC. To enhance statistical stability and reproducibility of the experimental results, we follow the experimental protocol of the QMIX, conducting five independent training runs with random initialization for each algorithm. The average test performance at each

time step is calculated, and the final results are reported as the mean performance across runs.

5 Experimental Results Analysis

5.1 Performance Analysis of CTMAPPO-Clip

To validate the effectiveness of the proposed CTMAPPO-Clip in multi-agent cooperative scenarios, we conducted a comparison of CTMAPPO-Clip with mainstream baseline algorithms across multiple scenarios in the SMAC benchmark environment. Our focus was on evaluating the algorithm's learning efficiency during training, policy convergence, and average win rate performance during the testing phase. Through comparative experiments, we provide a comprehensive assessment of CTMAPPO-Clip's advantages in addressing policy overfitting, enhancing generalization ability, and tackling the challenges posed by high-dimensional joint action spaces.

Table 3 Average Win Rates of MARL Algorithms on SMAC in Hard and Super Hard Scenarios

Scenarios	Difficulty	CTMAPPO-Clip	QMIX	MAPPO
3s_vs_5z	Hard	100.00%	100.00%	98.12%
2c_vs_64zg	Hard	100.00%	100.00%	100.00%
5m_vs_6m	Hard	**85.54%**	78.61%	74.33%
8m_vs_9m	Hard	**94.27%**	92.02%	87.64%
MMM2	Super Hard	97.21%	**100.00%**	96.72%
3s5z_vs_3s6z	Super Hard	**76.14%**	72.52%	46.32%
6h_vs_8z	Super Hard	**86.18%**	81.57%	50.28%
27m_vs_30m	Super Hard	**96.47%**	90.07%	88.64%
corridor	Super Hard	98.78%	**100.00%**	0.00%

According to the experimental results on SMAC under Hard and Super Hard difficulty levels, CTMAPPO-Clip demonstrates superior average win rates in most scenarios, validating its effectiveness and robustness in multi-agent cooperative tasks. Compared to the fine-tuned QMIX algorithm and the original MAPPO algorithm, CTMAPPO-Clip shows stronger adaptability and better strategic performance in several key scenarios.

In all hard difficulty scenarios, CTMAPPO-Clip performs comparably or even better than the fine-tuned QMIX. In the classic asymmetric competitive scenarios 3s_vs_5z and 2c_vs_64zg, CTMAPPO-Clip and QMIX both achieve 100.00% win rates, while MAPPO slightly lags behind in 3s_vs_5z with a win rate of only 98.12%. In the more complex balanced battle scenarios 5m_vs_6m and 8m_vs_9m, CTMAPPO-Clip outperforms QMIX (78.61%, 92.02%) and MAPPO (74.33%, 87.64%) with win rates of 85.54% and 94.27%, respectively. This demonstrates that CTMAPPO-Clip offers higher stability and flexibility, especially in scenarios requiring dynamic strategy adjustments with relatively balanced adversaries.

In the Super Hard scenarios, CTMAPPO-Clip's advantages become even more pronounced. In scenarios requiring high cooperation, such as 3s5z_vs_3s6z and 6h_vs_8z, CTMAPPO-Clip achieves win rates of 76% and 86%, significantly outperforming MAPPO (46.32%, 50.28%) and QMIX (72.52%, 81.57%). This indicates that CTMAPPO-Clip is better at coordinating strategies between multiple agents to accomplish complex task objectives. This advantage is largely attributed to the introduction of the Transformer architecture, which uses self-attention mechanisms to capture collaboration dependencies among agents, enabling the policy network to learn richer joint policy representations. Additionally, the advantage clipping mechanism introduced during training helps mitigate instability caused by policy overfitting in traditional MAPPO. By limiting excessively high advantage values, the algorithm becomes more stable during gradient updates, preventing the system from getting trapped in local optima.

However, it is also observed that in certain scenarios, such as MMM2 and corridor, the fine-tuned QMIX slightly outperforms or performs equally to CTMAPPO-Clip. Especially in the corridor scenario, where MAPPO fails completely (with a win rate of 0.00%), QMIX still achieves 100.00%. These types of scenarios usually have strong regularities or decision-making characteristics sensitive to state-action value functions (Q-values). The value-based cooperative mechanism used by QMIX allows for more direct global Q-value aggregation, which may provide an advantage in highly structured battles.

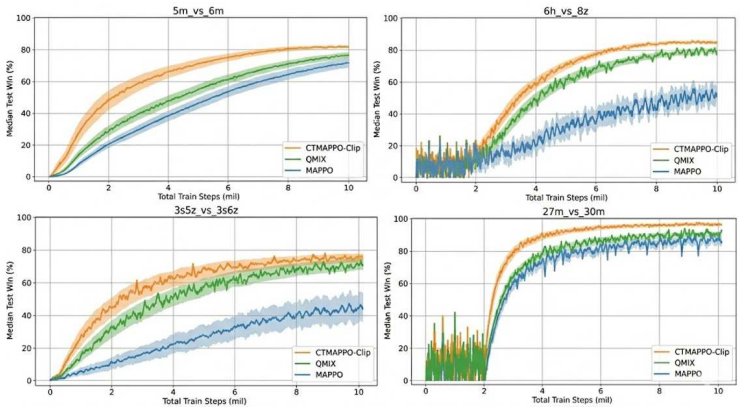


Fig. 2 Comparison of Average Test Win Rates between CTMAPPO-Clip and Baseline Algorithms on SMAC (Hard Scenarios) (a) 3s5z_vs_3s6z (b) 5m_vs_6m (c) 6h_vs_8z (d) 27m_vs_30m

Figure 2 presents changes in average test win rates during training for CTMAPPO-Clip, QMIX, and MAPPO across four SMAC Hard scenarios: 3s5z_vs_3s6z, 5m_vs_6m, 6h_vs_8z, and 27m_vs_30m. It can be observed that CTMAPPO-Clip (orange curve) demonstrates faster convergence and higher final win rates in most scenarios, particularly in 6h_vs_8z and 5m_vs_6m, where it significantly outperforms the fine-tuned QMIX and MAPPO, showcasing stronger policy generalization and stability. In the relatively balanced 27m_vs_30m scenario, CTMAPPO-Clip performs similarly to QMIX,

but MAPPO falls behind. This further validates the effectiveness of the proposed advantage clipping mechanism and Transformer architecture in enhancing multi-agent collaboration and addressing policy overfitting issues.

5.2 Ablation Experiments

To assess the contribution of key modules in CTMAPPO-Clip, we conducted ablation experiments focusing on the Transformer policy architecture and the advantage clipping mechanism. Experiments were conducted on two representative SMAC super hard scenarios: 6h_vs_8z and MMM2. To isolate the effect of each module, we removed them individually to create three ablation variants.

(1) CTMAPPO-Clip w/o Clipping: removes the advantage clipping mechanism and uses raw advantage values for policy updates;

(2) CTMAPPO-Clip w/o Transformer: replaces the Transformer with a standard two-layer MLP;

(3) CTMAPPO-Clip Full: the complete version with both the advantage clipping mechanism and Transformer module.

The results of the ablation experiments are shown in Figure 3.

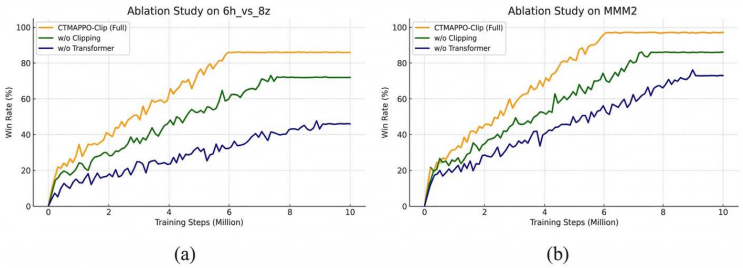


Fig. 3 Ablation Experiment Results (a) 6h_vs_8z (b) MMM2

In the 6h_vs_8z scenario, the full version of CTMAPPO-Clip achieved an average win rate of 86.35%, significantly outperforming the variants where the advantage clipping mechanism was removed (72.63%) or replaced with a standard MLP network (46.85%). This result demonstrates that in highly imbalanced and intense combat tasks, the Transformer architecture is better at modeling the collaborative dependencies between agents, preventing fragmented decisions based on local information. At the same time, the advantage clipping mechanism effectively mitigates policy overfitting during training, allowing agents to more robustly handle complex adversarial situations.

In the MMM2 scenario, the full CTMAPPO-Clip algorithm achieved a high win rate of 97.27%, while the version without the advantage clipping mechanism dropped to 86.31%, and the version without the Transformer architecture reached only 73.15%. The MMM2 scenario involves multi-unit coordination and multi-objective task management, placing higher demands on policy modeling capability and robustness. The experimental results further validate that the Transformer architecture, through its self-attention mechanism, captures the global dependencies among multiple agents,

effectively enhancing the policy's expressive power in complex decision spaces. Additionally, the advantage clipping mechanism prevents policy collapse or instability caused by outlier advantage values in the later stages of training.

Table 4 Comparison of Average Win Rates and Convergence Speed in 6h_vs_8z Scenario (Ablation Study)

Algorithm	Win Rate(%)	Convergence Speed(mil)
CTMAPPO-Clip(Full)	86.35	6.0
w/o Clipping	72.63	7.5
w/o Transformer	46.85	9.1

Table 5 Comparison of Average Win Rates and Convergence Speed in MMM2 Scenario (Ablation Study)

Algorithm	Win Rate(%)	Convergence Speed(mil)
CTMAPPO-Clip(Full)	97.27	6.1
w/o Clipping	86.31	7.7
w/o Transformer	73.15	9.2

As shown in Table 3 and Table 4, the full CTMAPPO-Clip algorithm demonstrates faster convergence in both the 6h_vs_8z and MMM2 scenarios, further validating the advantages of the proposed method in terms of training efficiency. Specifically, in the 6h_vs_8z scenario, CTMAPPO-Clip(Full) converges in approximately 6.0 million steps, while removing the advantage clipping mechanism (w/o Clipping) slows convergence to 7.5 million steps. Further removing the Transformer architecture from the policy network (w/o Transformer) significantly slows the training process, requiring 9.1 million steps to converge. A similar trend is observed in the MMM2 scenario: CTMAPPO-Clip(Full) converges in 6.1 million steps, while its two variants converge in 7.7 million steps and 9.2 million steps, respectively.

These results indicate that the Transformer architecture, with its powerful sequence modeling capability and self-attention mechanism for expressing inter-agent dependencies, enables the policy network to better capture key collaborative features, thus accelerating the learning of effective policies. Meanwhile, the advantage clipping mechanism effectively suppresses excessively high advantage values during training, alleviating the drastic fluctuations in the policy update process, thereby improving policy optimization stability and training efficiency. The synergy between these two components significantly enhances the model's learning speed and convergence stability in multi-agent scenarios, highlighting the significant potential of CTMAPPO-Clip in improving the training efficiency of multi-agent reinforcement learning.

Additionally, we conducted an experimental analysis of the advantage clipping hyperparameter in CTMAPPO-Clip.

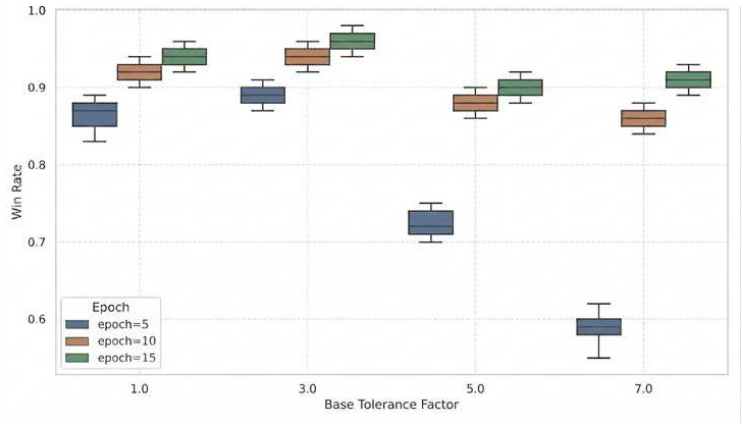


Fig. 4 Win Rate Performance of CTMAPPO-Clip under Different Advantage Clipping Parameters and Epoch Settings in the MMM2 Scenario

Figure 4 shows the win rate performance of CTMAPPO-Clip in the MMM2 scenario under different advantage clipping parameter settings (0.05, 0.1, 0.15, 0.2) and varying epoch values (5, 10, 15). The experimental results indicate that both the clipping parameter and the number of training epochs significantly affect the final performance. In general, moderate clipping parameter ranges such as 0.1 and 0.15 maintain higher average win rates and medians across different epoch settings, exhibiting better robustness and stability. On the other hand, smaller or larger clipping parameters, such as 0.05 and 0.2, show higher volatility, especially when the number of epochs is 5. In particular, when the parameter is set to 0.2, the win rate drops significantly and is accompanied by larger variance, indicating that the model becomes more susceptible to outlier advantage values, leading to reduced policy stability. As the number of epochs increases, the overall performance stabilizes, but the choice of clipping parameter still significantly impacts the final convergence performance. Considering both performance and stability, a clipping parameter of 0.1 combined with a moderate number of training epochs achieves the best performance in this scenario. This further validates the effectiveness of the advantage clipping mechanism in CTMAPPO-Clip for enhancing the generalization ability of multi-agent policies.

6 Conclusion

Overall, this paper addresses the issue of policy overfitting in multi-agent reinforcement learning by proposing the CTMAPPO-Clip algorithm within the CTCE framework. Experimental results on the SMAC benchmark demonstrate its superior performance over mainstream methods such as QMIX and MAPPO, effectively enhancing policy generalization and robustness.

While CTMAPPO-Clip performs well in the evaluated settings, several avenues remain open for future research. First, the current approach assumes full observability; extending it to partially or non-fully observable environments would improve its applicability in real-world scenarios. Second, although the Transformer architecture

offers strong representational capacity, its computational cost remains high. Future work could explore more efficient attention mechanisms or graph neural networks to improve training efficiency and scalability. Furthermore, incorporating mechanisms such as meta-learning and adaptive communication in heterogeneous multi-agent systems and dynamic task settings could further enhance policy transferability and generalization, representing promising directions for continued investigation.

References

[1] Han, Z., Zhou, R., Xu, C., Zeng, Y., Zhang, R.: Inss: An intelligent scheduling orchestrator for multi-gpu inference with spatio-temporal sharing. IEEE Transactions on Parallel and Distributed Systems: A Publication of the IEEE Computer Society (10), 35 (2024)

[2] Jiang, J., Yang, L., Zhang, L.: An overview of micro/nanorobot swarm control: From fundamental understanding to autonomy. IEEE/ASME Transactions on Mechatronics **30**

[3] Hu, S., Fang, Z., Deng, Y., Chen, X., Fang, Y.: Collaborative perception for connected and autonomous driving: Challenges, possible solutions and opportunities (2024)

[4] Li, Z., Ji, Q., Ling, X., Liu, Q.: A comprehensive review of multi-agent reinforcement learning in video games. IEEE Transactions on Games (2025)

[5] Bai, Y., Sugawara, T.: Locally centralized execution for less redundant computation in multi-agent cooperation (2024)

[6] Leroy, P., Pisane, J., Ernst, D.: Value-based ctde methods in symmetric two-team markov game: from cooperation to team competition. arXiv preprint arXiv:2211.11886 (2022)

[7] Qiao, T., Cui, P., Zhang, Y.: Hierarchical reinforcement learning based multi-agent collaborative control approach. In: 2023 42nd Chinese Control Conference (CCC), pp. 8645–8650 (2023). IEEE

[8] Venugopal, A., Milani, S., Fang, F., Ravindran, B.: Mabl: Bi-level latent-variable world model for sample-efficient multi-agent reinforcement learning. arXiv preprint arXiv:2304.06011 (2023)

[9] Jin, W., Du, H., Zhao, B., Tian, X., Shi, B., Yang, G.: A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives. arXiv preprint arXiv:2503.13415 (2025)

[10] Du, C., Fang, M., Li, Y., Li, L., Wang, X.: Smac, a mitochondrial protein that promotes cytochrome c–dependent caspase activation by eliminating iap inhibition. Cell **102**(1), 33–42 (2000)

[11] Wei, D., Zhang, L., Liu, Q., Chen, H., Huang, J.: Uav swarm cooperative dynamic target search: A mappo-based discrete optimal control method. Drones **8**(6), 214 (2024)

[12] Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., Wu, Y.: The surprising effectiveness of ppo in cooperative multi-agent games. Advances in neural information processing systems **35**, 24611–24624 (2022)

[13] Du, C., Fang, M., Li, Y., Li, L., Wang, X.: Smac, a mitochondrial protein that promotes cytochrome c–dependent caspase activation by eliminating iap inhibition. Cell **102**(1), 33–42 (2000)

[14] Manna, S., Loeffler, T.D., Batra, R., Banik, S., Chan, H., Varughese, B., Sasikumar, K., Sternberg, M., Peterka, T., Cherukara, M.J., *et al.*: Learning in continuous action space for developing high dimensional potential energy models. Nature communications **13**(1), 368 (2022)

[15] Hu, J., Hu, S., Liao, S.-w.: Policy regularization via noisy advantage values for cooperative multi-agent actor-critic methods. arXiv preprint arXiv:2106.14334 (2021)

[16] Papoudakis, G., Christianos, F., Rahman, A., Albrecht, S.V.: Dealing with non-stationarity in multi-agent deep reinforcement learning. arXiv preprint arXiv:1906.04737 (2019)

[17] Zhang, M., Tong, W., Zhu, G., Xu, X., Wu, E.Q.: Sqix: Qmix algorithm activated by general softmax operator for cooperative multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems **54**(11), 6550–6560 (2024)

[18] Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., et al.: Value-decomposition networks for cooperative multi-agent learning. arXiv preprint arXiv:1706.05296 (2017)

[19] Agarwal, S., Krueger, G., Clark, J., Radford, A., Kim, J.W., Brundage, M.: Evaluating clip: towards characterization of broader capabilities and downstream implications. arXiv preprint arXiv:2108.02818 (2021)

[20] De Witt, C.S., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P.H., Sun, M., Whiteson, S.: Is independent learning all you need in the starcraft multi-agent challenge? arXiv preprint arXiv:2011.09533 (2020)

[21] Xu, J., Sobhy, O., Alrifaee, B.: Xp-marl: Auxiliary prioritization in multi-agent reinforcement learning to address non-stationarity. arXiv preprint arXiv:2409.11852 (2024)

[22] Kang, H., Chang, X., Mišić, J., Mišić, V.B., Fan, J., Liu, Y.: Cooperative uav resource allocation and task offloading in hierarchical aerial computing systems: A mappo-based approach. IEEE Internet of Things Journal **10**(12), 10497–10509 (2023)

[23] Li, Z., Zuo, X., Song, Y., Liang, D., Xie, Z.: A multi-agent reinforcement learning based approach for automatic filter pruning. Scientific Reports **14**(1), 31193 (2024)

[24] Zhao, L.-y., Chang, T.-q., Guo, L.-b., Zhang, J., Zhang, L., Ma, J.-d.: An overestimation reduction method based on the multi-step weighted double estimation using value-decomposition multi-agent reinforcement learning. Neural Processing Letters **56**(3), 152 (2024)

[25] Pan, L., Rashid, T., Peng, B., Huang, L., Whiteson, S.: Regularized softmax deep multi-agent q-learning. Advances in Neural Information Processing Systems **34**, 1365–1377 (2021)

[26] Xu, H., Xuan, J., Zhang, G., Lu, J.: Trust region policy optimization via entropy regularization for kullback–leibler divergence constraint. Neurocomputing **589**, 127716 (2024)

[27] Bai, H., Cheng, R.: Generalized population-based training for hyperparameter optimization in reinforcement learning. IEEE Transactions on Emerging Topics in Computational Intelligence **8**(5), 3450–3462 (2024)

[28] Feng, Z., Wu, D., Huang, M., Yuen, C.: Event-driven transformer-based reinforcement learning for trajectory design and channel assignment in multi-uav assisted communication. IEEE Transactions on Cognitive Communications and Networking (2025)

[29] Hu, S., Shen, L., Zhang, Y., Chen, Y., Tao, D.: On transforming reinforcement learning with transformers: The development trajectory. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(12), 8580–8599 (2024)

[30] Ma, X., Hou, M., Zhan, J., Zhong, R.: Enhancing production prediction in shale gas reservoirs using a hybrid gated recurrent unit and multilayer perceptron (gru-mlp) model. Applied Sciences **13**(17), 9827 (2023)