

HCPO: Hierarchical Conductor-Based Policy Optimization in Multi-Agent Reinforcement Learning

Zejiao Liu^{1*}, Junqi Tu^{2*}, Yitian Hong^{2*}, Luolin Xiong², Yaochu Jin^{3†}, Yang Tang^{2†}, Fangfei Li^{1†}

¹The School of Mathematics, East China University of Science and Technology, Shanghai, China

²The Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai, China

³The School of Engineering, Westlake University, Hangzhou, China

{liuzejiao, 23012389, y20200105}@mail.ecust.edu.cn, xiongluolin@gmail.com, jinyaochu@westlake.edu.cn, {yangtang, lifangfei}@ecust.edu.cn

Abstract

In cooperative Multi-Agent Reinforcement Learning (MARL), efficient exploration is crucial for optimizing the performance of joint policy. However, existing methods often update joint policies via independent agent exploration, without coordination among agents, which inherently constrains the expressive capacity and exploration of joint policies. To address this issue, we propose a conductor-based joint policy framework that directly enhances the expressive capacity of joint policies and coordinates exploration. In addition, we develop a Hierarchical Conductor-based Policy Optimization (HCPO) algorithm that instructs policy updates for the conductor and agents in a direction aligned with performance improvement. A rigorous theoretical guarantee further establishes the monotonicity of the joint policy optimization process. By deploying local conductors, HCPO retains centralized training benefits while eliminating inter-agent communication during execution. Finally, we evaluate HCPO on three challenging benchmarks: StarCraftII Multi-agent Challenge, Multi-agent MuJoCo, and Multi-agent Particle Environment. The results indicate that HCPO outperforms competitive MARL baselines regarding cooperative efficiency and stability.

Introduction

Cooperative Multi-Agent Reinforcement Learning (MARL) methods have driven significant progress across various fields, including autonomous driving (Chen et al. 2025), robot cooperative control (Gu et al. 2023), and smart grid (Zhang et al. 2022). However, the increasing number of agents in the environment leads to the exponential growth of the state space and the joint action space, which brings the scalability challenge in MARL. A widely adopted solution to tackle this challenge is the Centralized Training with Decentralized Execution (CTDE) paradigm (Feng et al. 2024; Na and Moon 2024). It updates agents' policies with global information during training, while ensuring that agents make

decisions only based on their own local information during execution. Typical CTDE algorithms such as MADDPG (Lowe et al. 2017), QMIX (Rashid et al. 2020), and MAPPO (Yu et al. 2022) attract widespread attention for their enhanced coordination and overall effectiveness.

Under the CTDE paradigm, efficient exploration is important in MARL (Zheng et al. 2021; Xu, Zhang, and Huang 2023; Zhang et al. 2023b). Since parameter sharing restricts the behavioral diversity among agents, consequently impairing their exploration capabilities and impeding task completion (Li, Pan, and Zhang 2024; Li and Zhu 2025), researchers have developed heterogeneous MARL algorithms (Kuba et al. 2022; Li, Pan, and Zhang 2024). Specifically, to tackle the non-stationarity problem arising from simultaneous agent decisions in heterogeneous settings, sequential update algorithms such as Heterogeneous-Agent Trust Region Policy Optimisation (HATRPO) (Kuba et al. 2022) and Agent-by-agent Policy Optimization (A2PO) (Wang et al. 2023) have been proposed. This technique allows subsequent agents to integrate the action and policy information of previous agents within each training iteration. However, most of the existing CTDE algorithms, such as HATRPO and A2PO, presume that joint policy is expressed as the product of individual policies. This limits the expressive capacity of joint policy, making it difficult for multi-agent systems to explore the optimal joint policy during training. Therefore, we propose a hierarchical joint policy framework to address the limitation in expression and enhance exploration for better performance.

Inspired by soccer training and competition, we propose a conductor-based method to enhance the ability of joint policy expression and exploration, as shown in Figure 1. A centralized conductor provides instructions to the entire team by considering the on-field status of both sides. Specifically, the team can be instructed to adopt offensive instructions such as high pressing, flank attack, or defensive instructions such as zonal marking and low block. Players make decisions according to the instruction of the conductor and their own observations. For example, after the conductor chooses and broadcasts the “high pressing” instruction to all blue

*These authors contributed equally.

†Corresponding authors.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

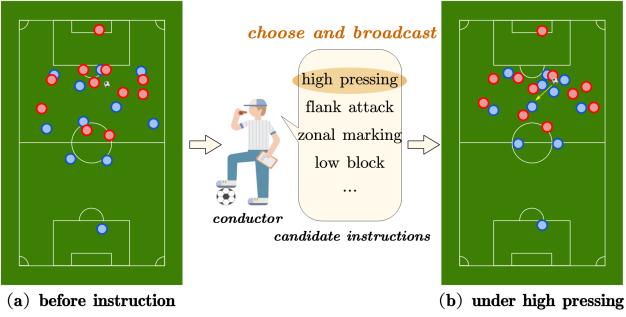


Figure 1: Visualization of conductor-based instructional impact in multi-agent learning: Blue and red dots represent players from opposing teams, with the blue team conductor providing strategic instructions.

teammates, each agent produces its own action by synthesizing this global instruction with its local observation. Consequently, some players drop back to consolidate defensive coverage, while others advance to compress space, jointly generating a coordinated pressing pattern.

In this work, we develop a heterogeneous sequential MARL framework, termed **Hierarchical Conductor-based Policy Optimization** (HCPO). Firstly, a joint policy model based on the conductor’s instructions is constructed for strong expressive capacity, which is inspired by a Gaussian mixture model. Subsequently, we establish a theoretically grounded update mechanism with strict monotonic improvement guarantees. By decomposing the joint policy optimization into conductor-level instruction and agent-level execution, we derive dual trust region constraints that ensure robust policy evolution. To enable practical decentralized execution, we further deploy local conductors to all agents and adapt their policies through the cross-entropy method with the centralized conductor, eliminating dependency on inter-agent communication. In addition, we perform extensive empirical validation across standardized benchmarks, including StarCraft II Multi-Agent Challenge (SMAC), Multi-agent MuJoCo (MA-MuJoCo), and Multi-agent Particle Environment (MPE) environments. The results show that HCPO outperforms existing competitive MARL algorithms. The main contributions are summarized as follows.

- **Hierarchical Conductor-based Policy Expression:** To enhance policy expressive capacity and guide multi-agent exploration, we propose a conductor-based joint policy framework: $\pi_{\text{mar}}(a|s) \triangleq \mathbb{E}_{M \sim w(\cdot|s)} \pi(a|s, M)$.
- **HCPO Algorithm and Monotonic Improvement Guarantee:** With the above conductor-based joint policy expression, we derive a new decomposition approach that breaks down the joint policy’s KL divergence into two components: the conductor policies’ KL divergence and the agent policies’ KL divergence. Then, we present a policy improvement inequality and design a two-level policy update mechanism for the conductor and agents. Finally, we prove that HCPO can ensure a monotonic improvement of the joint policy.

- **Extensive Experimental Validation:** Comprehensive evaluations on MARL benchmarks demonstrate the superior performance of HCPO over strong MARL baselines. The results show improvements in cooperative efficiency, policy stability, and exploration.

Related Work

Exploration: In MARL, the exponential expansion of joint state and action spaces as the number of agents increases severely challenges their ability to efficiently identify high-value states and actions (Liu et al. 2021b). To improve the performance of MARL algorithm, research on the exploration of state and policy spaces is crucial. In terms of state space exploration, the classical work Multi-Agent Variational Exploration (MAVEN) (Mahajan et al. 2019) guides multi-agent systems to learn diverse exploration patterns by maximizing the mutual information between agent trajectories and latent variables. To further improve exploration efficiency, methods like (Jo et al. 2024) and (Li and Zhu 2025) are proposed. Regarding policy space exploration, the current research focuses on constructing policy diversity incentive mechanisms (Dou et al. 2024). For example, the paper (Xu, Zhang, and Huang 2023) proposes an exploration method based on joint policy diversity for sparse-reward multi-agent tasks. It drives agents to explore new policies by maximizing the cross-entropy between the current joint policy and previous joint policies. However, most existing methods, either explicitly or implicitly, assume that the joint policy is the product of individual policies (Kuba et al. 2022; Dou et al. 2024; Jo et al. 2024). This decoupled method limits the expressive capacity of joint policy, and sometimes hampers agents’ exploration during training. To address this, we design a conductor-based framework that enhances the joint policy’s expressive capacity. Specifically, our method provides instructions (latent variables (Mahajan et al. 2019; Ibrahim and Fayad 2022)) to guide agents’ exploration in the policy space during training, enabling them to explore new and potentially high-value policies that traditional methods might overlook.

Hierarchical mechanism: In recent years, hierarchical mechanisms have been increasingly adopted (Vezhnevets et al. 2017; Ahilan and Dayan 2019; Paolo et al. 2025). MAVEN (Mahajan et al. 2019) embeds a latent space for hierarchical control within the CTDE framework, which alleviates the expressive limitations introduced by the monotonicity hypothesis in QMIX (Rashid et al. 2020). Through MAVEN, agents condition their behavior on the shared latent variable to enhance exploration and mitigate issues related to suboptimal policies. Furthermore, HAVEN (Xu et al. 2023) proposes a QMIX-style policy optimization framework with a dual coordination mechanism between layers and agents. Skill discovery represents another key direction in hierarchical MARL (He, Shao, and Ji 2020; Zhang et al. 2023a), enabling agents to autonomously learn diverse teamwork skills without requiring manually designed rewards (Liu et al. 2022, 2025). For example, hierarchical learning with skill discovery method (Yang, Borovikov, and Zha 2020) is a two-level MARL algorithm that uses latent skill variables and intrinsic rewards for unsupervised

skill discovery. The hierarchical multi-agent skill discovery (Yang et al. 2023) further extends the research on skill discovery by introducing team and individual skills. By employing the probabilistic graphical model, it formulates multi-agent skill discovery as an inference problem and leverages transformer structure to assign skills for coordination. To handle dynamic team composition, COPA (Liu et al. 2021a) proposes a coach-player hierarchy where a centralized coach periodically broadcasts strategies derived from global information. This design, however, retains the monotonicity constraint of QMIX and relies on communication during execution. Consequently, existing methods mainly use the monotonicity hypothesis imposed by QMIX and optimize hierarchical policies through variational inference. In contrast, our HCPO provides a theoretical guarantee of monotonic improvement in joint policy performance, without requiring any hypothesis regarding the joint action-value function. Additionally, our approach combines a conductor-based framework with trust region and sequential update methods, diverging from the mutual information and variational inference approaches commonly employed.

Background

Cooperative MARL Problem Formulation

In this paper, we consider fully cooperative multi-agent task as a Decentralized Markov Decision Process (DEC-MDP) (Bernstein et al. 2002; Kuba et al. 2022; Wang et al. 2023), which is usually modeled as a tuple $G = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$. Here, $\mathcal{N} = \{1, 2, \dots, N\}$ is the set of agents. \mathcal{S} and \mathcal{A} represent the state space of the environment and the whole joint action space, respectively. Each agent i takes action $a^i \in \mathcal{A}^i$, and $\mathcal{A} = \prod_{i=1}^N \mathcal{A}^i$. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability function. $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function shared by all agents and $\gamma \in [0, 1]$ is the discount factor. To improve coordinated exploration, we propose a hierarchical conductor-based framework that adapts agent policies with the instructions from the conductor. Specifically, let M denote the conductor’s instructional decision, sampled from a *centralized* instruction preference distribution $w(\cdot|s)$. The multi-agent joint policy π_{mar} is formulated as the expectation over all possible instructions M : $\pi_{\text{mar}}(\mathbf{a}|s) \triangleq \mathbb{E}_{M \sim w(\cdot|s)} \pi(\mathbf{a}|s, M)$, where $\mathbf{a} = (a^1, a^2, \dots, a^N) \in \mathcal{A}$ is the joint action. To facilitate decentralized execution, we equip each agent with an independent *local* conductor $w^i(\cdot|o^i)$, sharing the same instruction space as the *centralized* conductor. For clarity, the term “conductor” is primarily used to denote the centralized conductor unless otherwise specified. Hereafter, we assume that the conductor has K discrete instructions available to choose at each time. For any given instruction M , the corresponding instruction-conditional joint policy is defined as the product of individual agent policies conditioned on M , i.e. $\pi(\mathbf{a}|s, M) = \prod_{i=1}^N \pi^i(a^i|s, M)$. Therefore, under the conductor-based framework, our goal is to maximize the expected cumulative reward:

$$J(\pi_{\text{mar}}) \triangleq \mathbb{E}_{\mathfrak{s}^{0:\infty}, \mathfrak{m}_w^{0:\infty}, \mathfrak{a}_{\pi}^{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \mathbf{a}_t) \right]. \quad (1)$$

In the above equation, we denote “ $s_{0:\infty} \sim \rho_{\pi_{\text{mar}}}^{0:\infty}$ ” as “ $\mathfrak{s}_{\rho_{\pi_{\text{mar}}}^{0:\infty}}$ ”, “ $M_{0:\infty} \sim w_{0:\infty}$ ” as “ $\mathfrak{m}_w^{0:\infty}$ ”, and “ $a_{0:\infty} \sim \pi_{0:\infty}(M_{0:\infty})$ ” as “ $\mathfrak{a}_{\pi}^{0:\infty}$ ” for the sake of brevity. Hereafter, we use this notation wherever no ambiguity arises. Consequently, the state value function $V_{\pi_{\text{mar}}}(s)$ is defined as the expected cumulative return under the multi-agent joint policy π_{mar} :

$$V_{\pi_{\text{mar}}}(s) \triangleq \mathbb{E}_{\mathfrak{m}_w^{0:\infty}, \mathfrak{a}_{\pi}^{0:\infty}, \mathfrak{s}_{\rho_{\pi_{\text{mar}}}^{1:\infty}}} \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s \right]. \quad (2)$$

Similarly, we define the state-action value function $Q_{\pi_{\text{mar}}}(s, a) \triangleq \mathbb{E}_{\mathfrak{s}_{\rho_{\pi_{\text{mar}}}^{1:\infty}}, \mathfrak{m}_w^{1:\infty}, \mathfrak{a}_{\pi}^{1:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \mathbf{a}_0 = a \right]$. The joint advantage function is written as: $A_{\pi_{\text{mar}}}(s, a) \triangleq Q_{\pi_{\text{mar}}}(s, a) - V_{\pi_{\text{mar}}}(s)$.

Sequential Policy Update Mechanism

To address non-stationarity issues in MARL, the sequential update for agents has been widely investigated (Kuba et al. 2022; Wang et al. 2023; Dou et al. 2024; Wan et al. 2025). HATRPO extends the single-agent Trust Region Policy Optimization (TRPO) to multi-agent domain, utilizing multi-agent advantage decomposition to facilitate the sequential updates of agent policies. In HATRPO, each agent updates its policy parameters through the following protocol:

$$\theta_{k+1}^{i_n} = \arg \max_{\theta^{i_n}} \mathbb{E}_{s, \mathbf{a}^{i_{1:n-1}}, a^{i_n}} \left[A_{\pi_{\theta_k}}^{i_n}(s, \mathbf{a}^{i_{1:n-1}}, a^{i_n}) \right], \quad (3)$$

$$\text{subject to } \mathbb{E}_{s \sim \rho_{\pi_{\theta_k}}} \left[D_{\text{KL}} \left(\pi_{\theta_k^{i_n}}^{i_n}(\cdot|s), \pi_{\theta_{k+1}^{i_n}}^{i_n}(\cdot|s) \right) \right] \leq \delta. \quad (4)$$

In (3), the compact notation $\mathbb{E}_{s, \mathbf{a}^{i_{1:n-1}}, a^{i_n}}$ stands for the full expectation $\mathbb{E}_{s \sim \rho_{\pi_{\theta_k}}, \mathbf{a}^{i_{1:n-1}} \sim \pi_{\theta_{k+1}^{i_{1:n-1}}}^{i_{1:n-1}}, a^{i_n} \sim \pi_{\theta_k^{i_n}}}$. Here, $\theta_{k+1}^{i_n}$ denotes the policy parameter for agent i_n in episode $k+1$. $D_{\text{KL}}(\cdot, \cdot)$ is the KL-divergence between two policies, and δ is a hyperparameter. It is important to note that during the $k+1$ -th episode, policy $\theta_{k+1}^{i_n}$ leverages the updated policies $\pi_{\theta_{k+1}^{i_{1:n-1}}}^{i_{1:n-1}}$ from the preceding agents $i_{1:n-1}$. This is the core idea behind the sequential update mechanism.

Methods

Value Functions

In the previous section, we presented a conductor-based framework that enhances learning through instruction guidance. To evaluate this framework, we now need quantitative metrics to assess both the learning process and policy performance. First, we define the value function for the conductor’s instruction M as: $Q_{\pi_{\text{mar}}}(M|s) \triangleq \mathbb{E}_{\mathfrak{a}_{\pi}^{0:\infty}, \mathfrak{s}_{\rho_{\pi_{\text{mar}}}^{1:\infty}}, \mathfrak{m}_w^{1:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, M_0 = M \right]$. The instruction advantage function is written as:

$$A_{\pi_{\text{mar}}}(M|s) \triangleq Q_{\pi_{\text{mar}}}(M|s) - V_{\pi_{\text{mar}}}(s). \quad (5)$$

Here, $A_{\pi_{\text{mar}}}(M|s)$ measures the relative benefit of instruction M compared to all other instructions $M' \sim w(\cdot|s)$.

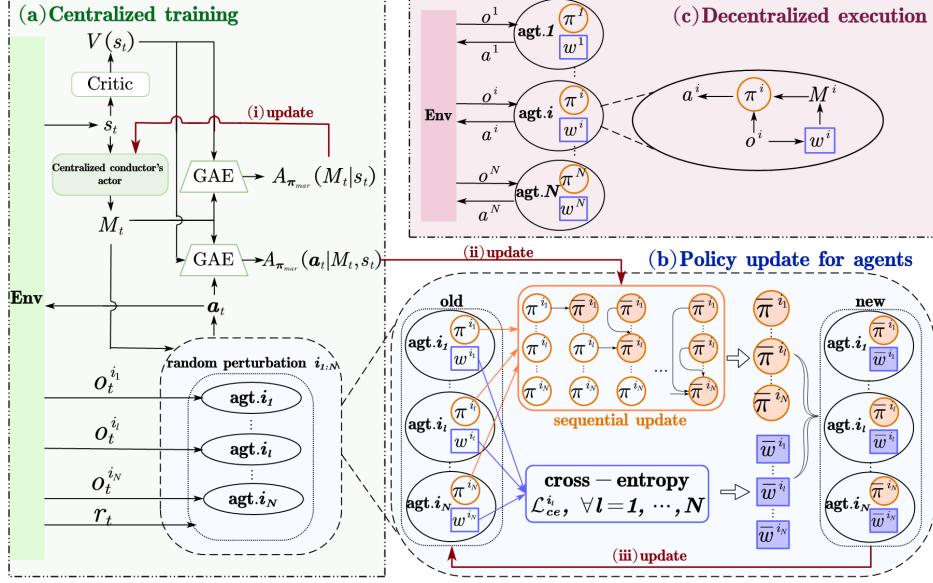


Figure 2: The overall framework of HCPO. (a) **Centralized training:** A two-level policy update mechanism with a virtual centralized conductor is proposed, leveraging well-designed advantage functions. (b) **Policy update for agents:** Here, local agents' policies denoted by orange ellipses are optimized through sequential updates, and local conductors' policies denoted by blue rectangles are optimized through the cross-entropy method. During this iteration, policies with shaded outlines represent updated versions, while those without shading indicate unmodified ones. (c) **Decentralized execution:** HCPO enables agents to make decisions based only on local information.

By choosing M to maximize $A_{\pi_{\text{mar}}}(M|s)$ with generalized advantage estimation (GAE) (Schulman et al. 2018), we optimize the conductor's policy w , as illustrated in Figure 2(i). Additionally, we define a joint action advantage function for agents' joint actions as follows:

$$A_{\pi_{\text{mar}}}(\mathbf{a}|s, M) \triangleq Q_{\pi_{\text{mar}}}(s, \mathbf{a}) - Q_{\pi_{\text{mar}}}(M|s). \quad (6)$$

The advantage function $A_{\pi_{\text{mar}}}(\mathbf{a}|s, M)$ evaluates the joint action \mathbf{a} over all other possible joint actions $\mathbf{a}' \sim \pi(\cdot|s, M)$. As illustrated in Figure 2(ii), maximizing this advantage function enables the optimization of the instruction-conditional joint policy $\pi(\cdot|s, M)$, thereby favoring actions that yield superior expected returns. Based on the above definitions, we can find that:

$$A_{\pi_{\text{mar}}}(s, \mathbf{a}) = A_{\pi_{\text{mar}}}(M|s) + A_{\pi_{\text{mar}}}(\mathbf{a}|s, M). \quad (7)$$

After the conductor chooses an instruction $M^j, j \in \{1, 2, \dots, K\}$, we update each individual agent's policy $\pi^{i_l}(a^{i_l}|s, M^j)$ in the order determined by the random perturbation set $i_{1:N} = \{i_1, i_2, \dots, i_N\}$. In addition, at the state s , based on the conductor's any instruction M^j , after the previous agents $i_{1:l}$ take joint action $\mathbf{a}^{i_{1:l}}$, we define the expected value of $\mathbf{a}^{i_{1:l}}$ as:

$$Q_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) \triangleq \mathbb{E}_{\mathbf{a}_0^{-i_{1:l}} \sim \pi_0^{-i_{1:l}}(\cdot|s, M^j), \mathbf{a}_0^{i_{1:\infty}}, \mathcal{M}_w^{1:\infty}, \mathbf{q}^{1:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, M_0 = M^j, \mathbf{a}_0^{i_{1:l}} = \mathbf{a}^{i_{1:l}} \right], \quad (8)$$

where $-i_{1:l}$ is the complement of $i_{1:l}$.

Lemma 1. For any instruction $M^j, j \in \{1, 2, \dots, K\}$ chosen by the conductor, the conditional Q -function for agents $i_{1:l}$ satisfies:

$$Q_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) = \mathbb{E}_{\mathbf{a}^{-i_{1:l}} \sim \pi^{-i_{1:l}}(\cdot|s, M^j)} [Q_{\pi_{\text{mar}}}(s, \mathbf{a})], \quad (9)$$

where $\mathbf{a} = (\mathbf{a}^{i_{1:l}}, \mathbf{a}^{-i_{1:l}})$.

The proof is proposed in Appendix A.1. This lemma provides the basis for the subsequent definitions of advantage functions, which are crucial for evaluating the relative advantages of specific actions compared to average actions. Specifically, at the state s , based on the conductor's any action M^j , after the agents $i_{1:l}$ first take joint action $\mathbf{a}^{i_{1:l}}$, and agents $-i_{1:l}$ take joint action $\mathbf{a}^{-i_{1:l}} \sim \pi^{-i_{1:l}}(\cdot|s, M^j)$, we define the advantage function for $\mathbf{a}^{i_{1:l}}$ as:

$$A_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) \triangleq Q_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) - Q_{\pi_{\text{mar}}}(M^j|s). \quad (10)$$

It is noted that when $l = 0$, we have $A_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) = 0$. Besides, at the state s , based on the conductor's any instruction M^j , for any individual agent i_l , we define the advantage of its individual action a^{i_l} over all actions $a^{i_l'} \sim \pi^{i_l'}(\cdot|s, M^j)$:

$$A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M^j) \triangleq Q_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) - Q_{\pi_{\text{mar}}}^{i_{1:l-1}}(\mathbf{a}^{i_{1:l-1}}|s, M^j). \quad (11)$$

In the subsequent part, we introduce the conductor-based multi-agent advantage function decomposition lemma. As shown in Figure 2(ii), the lemma is designed to facilitate the transition from updating the instruction-conditional

joint policy $\pi(\mathbf{a}|s, M^j)$ to updating individual agents' policies $\pi^{i_l}(\mathbf{a}^{i_l}|s, M^j), l \in \mathcal{N}$.

Lemma 2. (*Conditional Advantage Decomposition*) Consider a cooperative Markov game with a joint policy π_{mar} . For any state s , any instruction M^j , and any subset of agents $i_{1:n} = \{i_1, i_2, \dots, i_n\} \subseteq \mathcal{N}$, the following equation holds for all states s , joint actions $\mathbf{a}^{i_{1:n}}$, and $M^j \sim w$:

$$A_{\pi_{\text{mar}}}^{i_{1:n}}(\mathbf{a}^{i_{1:n}}|s, M^j) = \sum_{l=1}^n A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M^j). \quad (12)$$

The proof is proposed in Appendix A.1.

Quantifying Policy Updates for HCPO

TRPO (Schulman et al. 2015) is a reinforcement learning algorithm that improves learning stability by constraining policy update magnitude. In this section, we explore a conductor-based mechanism that incorporates TRPO to measure the difference in expected returns between the new policy and its predecessor. This analysis serves as the foundation for designing effective policy update algorithms.

Proposition 1. As defined in Equation (1), the relationship between the expected return of the new policy $\bar{\pi}_{\text{mar}}$ and the old policy π_{mar} is expressed as:

$$J(\bar{\pi}_{\text{mar}}) = J(\pi_{\text{mar}}) + \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[\sum_{t=0}^{\infty} \gamma^t A_{\pi_{\text{mar}}}(s_t, \mathbf{a}_t) \right], \quad (13)$$

where $\tau := (s_0, M_0, \mathbf{a}_0, s_1, M_1, \mathbf{a}_1, \dots)$.

The proof is proposed in Appendix A.1. Similar to HATRPO, we introduce the approximation function $L_{\pi_{\text{mar}}}(\bar{\pi}_{\text{mar}})$, which serves as an alternative objective function for the new policy's performance function $J(\bar{\pi}_{\text{mar}})$:

$$L_{\pi_{\text{mar}}}(\bar{\pi}_{\text{mar}}) = J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}, M \sim \bar{w}, \mathbf{a} \sim \bar{\pi}} [A_{\pi_{\text{mar}}}(s, \mathbf{a})], \quad (14)$$

where the compact notation $\mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}, M \sim \bar{w}, \mathbf{a} \sim \bar{\pi}}$ stands for the full expectation $\mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}, M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)}$. Therefore, we can derive the following theorem.

Theorem 1. Under the proposed conductor-based framework, a significant policy improvement inequality holds for the joint policy π_{mar} :

$$\begin{aligned} J(\bar{\pi}_{\text{mar}}) &\geq J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}} \left[\mathbb{E}_{M \sim \bar{w}(\cdot|s)} A_{\pi_{\text{mar}}}(M|s) \right. \\ &\quad - \text{CD}_{\text{KL}}^{\max}(w, \bar{w}) + \mathbb{E}_{M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M) \\ &\quad \left. - \max C \sum_{j=1}^K w(M^j|s) \text{D}_{\text{KL}}(\pi(\cdot|s, M^j), \bar{\pi}(\cdot|s, M^j)) \right], \end{aligned} \quad (15)$$

where $C = \frac{4\gamma \max_{s,a} |A_{\pi_{\text{mar}}}(s, a)|}{(1-\gamma)^2}$, $\text{D}_{\text{KL}}^{\max}(w_k, \bar{w}) = \max_s \text{D}_{\text{KL}}(w_k(\cdot|s), \bar{w}(\cdot|s))$.

For proof see Appendix A.2. The inequality quantifies the expected return difference between the new policy $\bar{\pi}_{\text{mar}}$ and the existing policy π_{mar} under the conductor-based mechanism. This finding offers precise guidance for the subsequent policy update process.

Guaranteed Monotonic Joint Policy Optimization

In this section, we formulate the policy update mechanisms for both the centralized conductor's policy and individual agents' policies. Our theoretical analysis aims to establish the monotonic improvement guarantee for the conductor-based joint policy π_{mar} through a two-level optimization framework. This guarantee constitutes a critical theoretical foundation, ensuring progressive performance improvement through successive policy iterations while validating the conductor's instructions. The specific two-level policy update mechanisms in episode $k+1$ are as follows:

(i) The conductor's policy $w(\cdot|s) = w_k(\cdot|s)$ is updated first according to the rule:

$$w_{k+1} = \arg \max_{\bar{w}} \left[\mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}, k}, M \sim \bar{w}} A_{\pi_{\text{mar}}, k}(M|s) \right. \\ \left. - \text{CD}_{\text{KL}}^{\max}(w_k, \bar{w}) \right]. \quad (16)$$

(ii) For each M^j where $j \in \{1, 2, \dots, K\}$, the agents sequentially update their policies in accordance with the order $i_{1:N}$, following the update rule:

$$\begin{aligned} \pi_{k+1}^{i_l}(\cdot|s, M^j) &= \arg \max_{\bar{\pi}^{i_l}(\cdot|s, M^j)} \left[w_{k+1}(M^j|s) L_{\pi_{\text{mar}}, k}^{i_{1:l}} \left(\pi_{k+1}^{i_{1:l-1}}, \bar{\pi}^{i_l}|s, M^j \right) \right. \\ &\quad \left. - \max C w_k(M^j|s) \text{D}_{\text{KL}}(\pi_k^{i_l}(\cdot|s, M^j), \bar{\pi}^{i_l}(\cdot|s, M^j)) \right], \end{aligned} \quad (17)$$

where $L_{\pi_{\text{mar}}, k}^{i_{1:l}} \left(\pi_{k+1}^{i_{1:l-1}}, \bar{\pi}^{i_l}|s, M^j \right) \triangleq \mathbb{E}_{\mathbf{a}^{i_{1:l-1}} \sim \pi_{k+1}^{i_{1:l-1}}, a^{i_l} \sim \bar{\pi}^{i_l}} \left[A_{\pi_{\text{mar}}, k}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M^j) \right]$. For proof see Appendix A.4. Equation (17) presents a policy update mechanism using w_k to regulate the update magnitude between the agents' new and existing policies. This approach optimizes the exploration-exploitation trade-off, helps avoid local optima, and enhances the adaptability and effectiveness of the policy update process. In HCPO's practical implementation, we employ the CTDE framework to mitigate the limitations posed by communication interference, as shown in Figure 2(c). During training, we incorporate a virtual *centralized* conductor w parameterized by Ψ , and each individual agent is equipped with a *local* conductor network w^i (parameterized by ψ^i) and an actor network π^i (parameterized by θ^i). To maintain the theoretical assumption that agents update their policies using only local observations and the broadcast instruction M , we employ a two-stage training protocol. First, we collect experience with the centralized conductor and optimize its policy parameter Ψ . Then, every agent's actor network θ^i is sequentially updated to improve the policy. Second, we fix Ψ and distill its policy into ψ^i via cross-entropy loss (Chen et al. 2024), enabling fully decentralized execution at evaluation. We summarize our HCPO as Algorithm 1 of Appendix B.

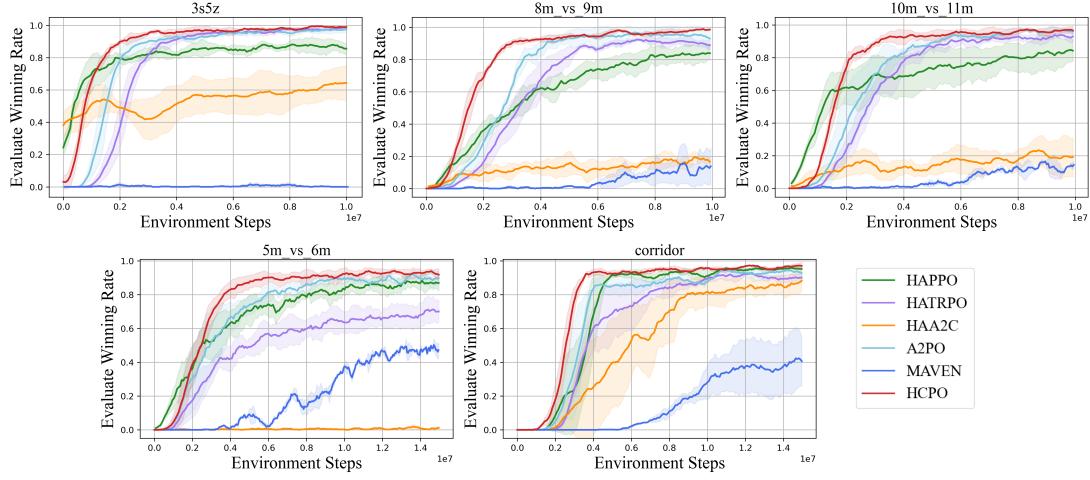


Figure 3: Performance comparison on SMAC. With the conductor-based joint policy enhancing learning efficiency, HCPO reliably outperforms all baselines.

Experiments

In this section, we evaluate HCPO on standard MARL benchmarks including SMAC (Samvelyan et al. 2019), MA-MuJoCo (Li, Pan, and Zhang 2024), and MPE (Lowe et al. 2017). It is compared against strong baselines: HATRPO (Kuba et al. 2022), HAPPO (Kuba et al. 2022), A2PO (Wang et al. 2023), HAA2C (Zhong et al. 2024). Detailed experimental setup and results are presented below.



Figure 4: Effective coordination in SMAC on the 3s5z map: A visual analysis of agent strategies.

Settings and Performance

SMAC: We evaluate algorithms on five SMAC maps, including the widespread hierarchical algorithm MAVEN (Mahajan et al. 2019). Each algorithm is tested with five different random seeds to ensure the robustness and reliability. The results demonstrate that our HCPO achieves outstanding performance in all test maps, as illustrated in Figure 3, with shadows showing standard deviation across different runs. Specifically, HCPO is the first to achieve a 90% winning rate on all maps, enhancing exploration and learning efficiency. Furthermore, HCPO exhibits the lowest standard deviation, demonstrating its high stability. In addition, we visualize the gameplay scenarios on the 3s5z map in Figure 4. Figure 4(a) shows the early game strategy where our team uses one stalker to draw enemy fire, while two other stalkers attack from

behind the zealots. Figure 4(b) presents the overall force division into two groups, with two stalkers providing support. These coordinated actions highlight the agents' effective collaboration, driven by the latent instructions of HCPO, which ultimately leads to victory. Detailed analysis and the evaluation method are presented in Appendix C.1.

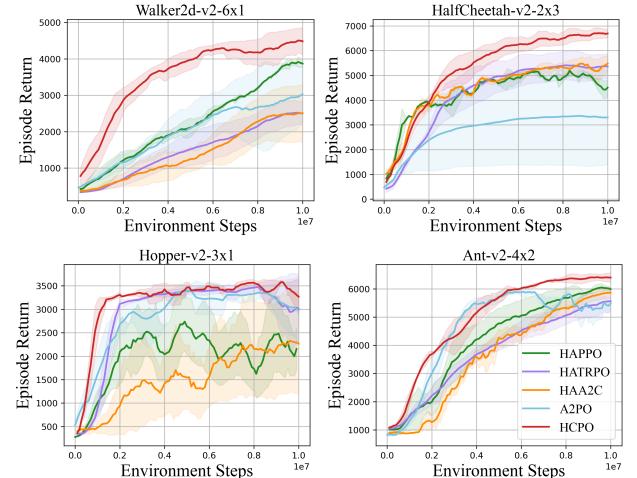


Figure 5: Comparative evaluation on MA-MuJoCo.

MA-MuJoCo: We compare HCPO with four advanced on-policy MARL algorithms using three different random seeds. As shown in Figure 5, HCPO not only achieves significantly higher final returns but also exhibits a lower standard deviation, indicating superior exploration and stability. Specifically, in *HalfCheetah-v2-2×3*, HCPO achieves around 23.42% higher final returns than the next best algorithm HAA2C. In Figure 6, we employ t-SNE (t-Distributed Stochastic Neighbor Embedding) technique to project the states explored by HCPO, HATRPO, and A2PO algorithms during the early stage of training in the *Walker2d-v2-6×1*

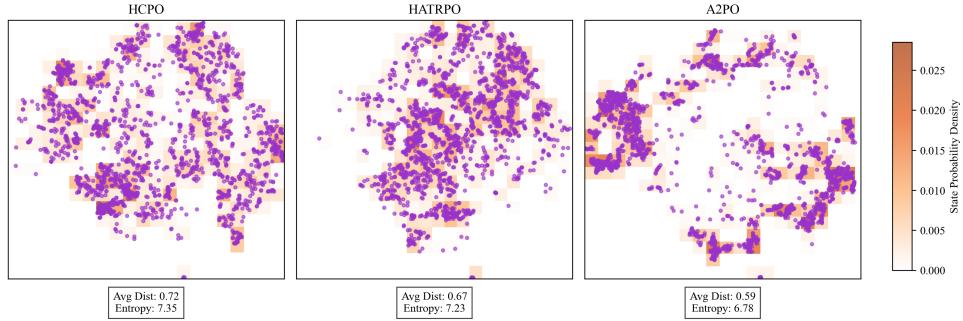


Figure 6: Exploration comparison: t-SNE visualization and entropy analysis in *Walker2d-v2-6×1*.

task onto a 2-D plane. Through entropy analysis and calculation of the average nearest neighbor distance, we can observe that HCPO demonstrates superior exploration. Detailed analysis is presented in Appendix C.2.

MPE: The results on MPE under three random seeds are shown in Figure 7. HCPO exhibits rapid policy improvement in the early stage (0-2 million steps) of training, indicating that HCPO has a high cooperative efficiency and sufficient exploration. Furthermore, compared with HATRPO and A2PO, HCPO shows significant stability and robustness. More experiments are illustrated in Appendix C.3.

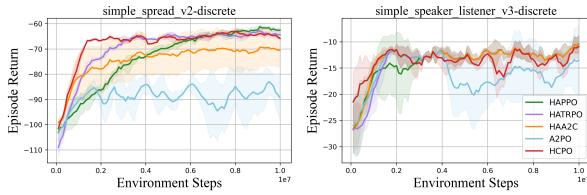


Figure 7: Performance comparison of HCPO and other strong MARL algorithms across different MPE tasks.

Ablation Studies

For several key components of HCPO, we conduct ablation studies with results shown in Figure 8. Figure 8(a) examines the impact of the conductor and varying the number of instructions (hyperparameter K) on performance. HCPO with the conductor shows a faster increase in winning rate and a higher final rate than without any conductor, demonstrating its effectiveness in boosting cooperation efficiency. The performance is also influenced by K . It is important to balance performance and resource consumption when selecting K . In Figure 8(b), we examine the hyperparameter δ_1 , which represents the KL-divergence constraint during the conductor's policy update. In Figure 8(c), we evaluate HCPO under four conductor configurations: a centralized conductor, a random conductor (non-learning baseline), no conductor and local conductors (the core of our HCPO algorithm). Figure 8(d) presents the final episode returns in a boxplot format. The results show that HCPO with local conductors achieves a median return comparable to that of a centralized conductor, while substantially outperforming

the variant without any conductor. Furthermore, replacing the learned instruction preference distribution with a non-learning conductor that outputs uniformly random instructions leads to inferior performance. These findings validate the effectiveness of our proposed update mechanisms. We present detailed analysis in Appendix C.4.

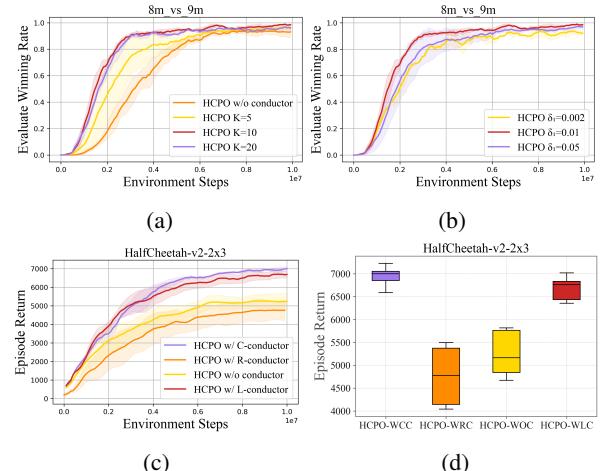


Figure 8: Ablation studies.

Conclusion and Future Work

In this paper, we introduce HCPO, a hierarchical MARL algorithm designed to enhance the expressive capacity of joint policies and improve exploration. By leveraging specific advantage functions, we propose a two-level policy update mechanism with monotonic improvement guarantees without a monotonicity hypothesis in QMIX. Based on the cross-entropy method, each agent is equipped with a local conductor. This not only improves cooperation but also avoids the limitations imposed by communication constraints. Through comprehensive experiments on SMAC, MA-MuJoCo, and MPE, HCPO demonstrates superior performance compared to competitive algorithms. The main limitation of this work is that we design an on-policy algorithm for our hierarchical framework. In the future, we plan to integrate the conductor-based mechanism of HCPO into off-policy algorithms to improve sample efficiency.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grants 62233005, U2441245, 62573198, 62173142, and in part by the Shanghai Institute for Mathematics and Interdisciplinary Sciences (SIMIS) under Grant SIMIS-ID-2025-SP.

References

- Ahilan, S.; and Dayan, P. 2019. Feudal Multi-Agent Hierarchies for Cooperative Reinforcement Learning. *arXiv:1901.08492*.
- Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberman, S. 2002. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*, 27(4): 819–840.
- Chen, X.; Wang, X.; Zhao, W.; Wang, C.; Cheng, S.; and Luan, Z. 2025. Hierarchical Deep Reinforcement Learning based Multi-Agent Game Control for Energy Consumption and Traffic Efficiency Improving of Autonomous Vehicles. *Energy*, 323: 135669.
- Chen, Y.; Mao, H.; Mao, J.; Wu, S.; Zhang, T.; Zhang, B.; Yang, W.; and Chang, H. 2024. PTDE: Personalized Training with Distilled Execution for Multi-Agent Reinforcement Learning. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI-24)*, 31–39. Jeju, South Korea: IJCAI Organization.
- Dou, H.; Dang, L.; Luan, Z.; and Chen, B. 2024. Measuring Mutual Policy Divergence for Multi-Agent Sequential Exploration. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*, volume 37, 76265–76288.
- Feng, P.; Liang, J.; Wang, S.; Yu, X.; Ji, X.; Chen, Y.; Zhang, K.; Shi, R.; and Wu, W. 2024. Hierarchical Consensus-Based Multi-Agent Reinforcement Learning for Multi-Robot Cooperation Tasks. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 642–649. Abu Dhabi, United Arab Emirates: IEEE.
- Gu, S.; Kuba, J. G.; Chen, Y.; Du, Y.; Yang, L.; Knoll, A.; and Yang, Y. 2023. Safe Multi-Agent Reinforcement Learning for Multi-Robot Control. *Artificial Intelligence*, 319: 103905.
- He, S.; Shao, J.; and Ji, X. 2020. Skill Discovery of Coordination in Multi-agent Reinforcement Learning. *arXiv:2006.04021*.
- Ibrahim, M.; and Fayad, A. 2022. Hierarchical Strategies for Cooperative Multi-Agent Reinforcement Learning. *arXiv:2212.07397*.
- Jo, Y.; Lee, S.; Yeom, J.; and Han, S. 2024. FoX: Formation-Aware Exploration in Multi-Agent Reinforcement Learning. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, volume 38, 12985–12994. Vancouver, Canada: AAAI Press.
- Kuba, J. G.; Chen, R.; Wen, M.; Wen, Y.; Sun, F.; Wang, J.; and Yang, Y. 2022. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, 1046. Virtual Only.
- Li, T.; and Zhu, K. 2025. Toward Efficient Multi-Agent Exploration with Trajectory Entropy Maximization. In *Proceedings of the 13th International Conference on Learning Representations (ICLR 2025)*. Singapore.
- Li, X.; Pan, L.; and Zhang, J. 2024. Kaleidoscope: Learnable Masks for Heterogeneous Multi-agent Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*.
- Liu, B.; Liu, Q.; Stone, P.; Garg, A.; Zhu, Y.; and Anandkumar, A. 2021a. Coach-Player Multi-Agent Reinforcement Learning for Dynamic Team Composition. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, 6860–6870. Virtual Only: PMLR.
- Liu, I.-J.; Jain, U.; Yeh, R. A.; and Schwing, A. 2021b. Cooperative Exploration for Multi-Agent Deep Reinforcement Learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, 6826–6836. Virtual Only: PMLR.
- Liu, S.; Shu, Y.; Guo, C.; and Yang, B. 2025. Learning Generalizable Skills from Offline Multi-Task Data for Multi-Agent Cooperation. In *Proceedings of the 13th International Conference on Learning Representations (ICLR 2025)*. Singapore.
- Liu, Y.; Li, Y.; Xu, X.; Dou, Y.; and Liu, D. 2022. Heterogeneous Skill Learning for Multi-Agent Tasks. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, volume 35, 37011–37023.
- Lowe, R.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, volume 30, 6379–6390.
- Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. MAVEN: Multi-Agent Variational Exploration. In *Advances in Neural Information Processing Systems (NeurIPS 2019)*, volume 32.
- Na, H.; and Moon, I.-C. 2024. LAGMA: Latent Goal-Guided Multi-Agent Reinforcement Learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, 37122–37140. Vienna, Austria: PMLR.
- Paolo, G.; Benechehab, A.; Cherkaoui, H.; Thomas, A.; and Kégl, B. 2025. TAG: A Decentralized Framework for Multi-Agent Hierarchical Reinforcement Learning. *arXiv:2502.15425*.
- Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *Journal of Machine Learning Research*, 21(178): 1–51.
- Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AA-MAS'19)*, 2186–2188. Montreal, Canada: IFAAMAS.
- Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust Region Policy Optimization. In *Proceedings*

- of the 32nd International Conference on Machine Learning (ICML 2015)*, 1889–1897. Lille, France: PMLR.
- Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2018. High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv:1506.02438.
- Vezhnevets, A. S.; Osindero, S.; Schaul, T.; Heess, N.; Jaderberg, M.; Silver, D.; and Kavukcuoglu, K. 2017. Feudal Networks for Hierarchical Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, 3540–3549. Sydney, Australia: PMLR.
- Wan, X.; Yang, C.; Yang, C.; Song, J.; and Sun, M. 2025. SrSv: Integrating Sequential Rollouts with Sequential Value Estimation for Multi-agent Reinforcement Learning. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, volume 39, 23333–23342. Philadelphia, Pennsylvania: AAAI Press.
- Wang, X.; Tian, Z.; Wan, Z.; Wen, Y.; Wang, J.; and Zhang, W. 2023. Order Matters: Agent-by-agent Policy Optimization. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. Kigali, Rwanda.
- Xu, P.; Zhang, J.; and Huang, K. 2023. Exploration via Joint Policy Diversity for Sparse-Reward Multi-Agent Tasks. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI-23)*, 326–334. Macao, China: IJCAI Organization.
- Xu, Z.; Bai, Y.; Zhang, B.; Li, D.; and Fan, G. 2023. HAVEN: Hierarchical Cooperative Multi-Agent Reinforcement Learning with Dual Coordination Mechanism. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, volume 37, 11735–11743. Washington, USA: AAAI Press.
- Yang, J.; Borovikov, I.; and Zha, H. 2020. Hierarchical Cooperative Multi-Agent Reinforcement Learning with Skill Discovery. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS’20)*. Virtual Only: IFAAMAS.
- Yang, M.; Yang, Y.; Lu, Z.; Zhou, W.; and Li, H. 2023. Hierarchical Multi-Agent Skill Discovery. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*, volume 36, 61759–61776.
- Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, volume 35, 24611–24624.
- Zhang, F.; Jia, C.; Li, Y.-C.; Yuan, L.; Yu, Y.; and Zhang, Z. 2023a. Discovering Generalizable Multi-Agent Coordination Skills from Multi-Task Offline Data. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. Kigali, Rwanda.
- Zhang, S.; Cao, J.; Yuan, L.; Yu, Y.; and Zhan, D.-C. 2023b. Self-Motivated Multi-Agent Exploration. In *Proceedings of the 22nd International Conference on Autonomous Agents and MultiAgent Systems (AAMAS’23)*, 476–484. London, United Kingdom: IFAAMAS.
- Zhang, Y.; Yang, Q.; An, D.; Li, D.; and Wu, Z. 2022. Multistep Multiagent Reinforcement Learning for Optimal Energy Schedule Strategy of Charging Stations in Smart Grid. *IEEE Transactions on Cybernetics*, 53(7): 4292–4305.
- Zheng, L.; Chen, J.; Wang, J.; He, J.; Hu, Y.; Chen, Y.; Fan, C.; Gao, Y.; and Zhang, C. 2021. Episodic Multi-Agent Reinforcement Learning with Curiosity-Driven Exploration. In *Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, 3757–3769.
- Zhong, Y.; Kuba, J. G.; Feng, X.; Hu, S.; Ji, J.; and Yang, Y. 2024. Heterogeneous-Agent Reinforcement Learning. *Journal of Machine Learning Research*, 25(32): 1–67.

Appendix

A. Additional Details for HCPO Framework

A.1 Useful Lemmas

Lemma 1. For any instruction $M^j, j \in \{1, 2, \dots, K\}$ chosen by the conductor, the conditional Q -function for agents $i_{1:l}$ satisfies:

$$Q_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) = \mathbb{E}_{\mathbf{a}^{-i_{1:l}} \sim \pi_0^{-i_{1:l}}(\cdot|s, M^j)} [Q_{\pi_{\text{mar}}}(s, \mathbf{a})], \quad (18)$$

where $\mathbf{a} = (\mathbf{a}^{i_{1:l}}, \mathbf{a}^{-i_{1:l}})$.

Proof.

$$\begin{aligned} Q_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) &= \mathbb{E}_{\mathbf{a}_0^{-i_{1:l}} \sim \pi_0^{-i_{1:l}}(\cdot|s, M^j)} \left[\mathbb{E}_{\mathbf{s}_{\rho_{\pi_{\text{mar}}}}^{1:\infty}, \mathfrak{M}_w^{1:\infty}, \mathbf{a}_{\pi}^{1:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, M_0 = M^j, \mathbf{a}_0^{i_{1:l}} = \mathbf{a}^{i_{1:l}}, \mathbf{a}_0^{-i_{1:l}} \right] \right] \\ &= \mathbb{E}_{\mathbf{a}_0^{-i_{1:l}} \sim \pi_0^{-i_{1:l}}(\cdot|s, M^j)} \left[\mathbb{E}_{\mathbf{s}_{\rho_{\pi_{\text{mar}}}}^{1:\infty}, \mathfrak{M}_w^{1:\infty}, \mathbf{a}_{\pi}^{1:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \mathbf{a}_0^{i_{1:l}} = \mathbf{a}^{i_{1:l}}, \mathbf{a}_0^{-i_{1:l}} \right] \right] \\ &= \mathbb{E}_{\mathbf{a}^{-i_{1:l}} \sim \pi_0^{-i_{1:l}}(\cdot|s, M^j)} [Q_{\pi_{\text{mar}}}(s, \mathbf{a})]. \end{aligned} \quad (19)$$

□

Lemma 2. (Conditional Advantage Decomposition) Consider a cooperative Markov game with a joint policy π_{mar} . For any state s , any instruction M^j , and any subset of agents $i_{1:n} = \{i_1, i_2, \dots, i_n\} \subseteq \mathcal{N}$, the following equation holds for all states s , joint actions $\mathbf{a}^{i_{1:n}}$, and $M^j \sim w$:

$$A_{\pi_{\text{mar}}}^{i_{1:n}}(\mathbf{a}^{i_{1:n}}|s, M^j) = \sum_{l=1}^n A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M^j). \quad (20)$$

Proof. According to the definition of the advantage function as given in (11), we proceed with the right-hand side as follows:

$$\begin{aligned} \sum_{l=1}^n A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M^j) &= \sum_{l=1}^n \left[Q_{\pi_{\text{mar}}}^{i_{1:l}}(\mathbf{a}^{i_{1:l}}|s, M^j) - Q_{\pi_{\text{mar}}}^{i_{1:l-1}}(\mathbf{a}^{i_{1:l-1}}|s, M^j) \right] \\ &= Q_{\pi_{\text{mar}}}^{i_{1:n}}(\mathbf{a}^{i_{1:n}}|s, M^j) - Q_{\pi_{\text{mar}}}(M^j|s) \\ &= A_{\pi_{\text{mar}}}^{i_{1:n}}(\mathbf{a}^{i_{1:n}}|s, M^j). \end{aligned} \quad (21)$$

□

Proposition 1. As defined in (1), the relationship between the expected return of the new policy $\bar{\pi}_{\text{mar}}$ and the old policy π_{mar} is expressed as:

$$J(\bar{\pi}_{\text{mar}}) = J(\pi_{\text{mar}}) + \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[\sum_{t=0}^{\infty} \gamma^t A_{\pi_{\text{mar}}}(s_t, \mathbf{a}_t) \right], \quad (22)$$

where $\tau := (s_0, M_0, \mathbf{a}_0, s_1, M_1, \mathbf{a}_1, \dots)$.

Proof. We begin by expanding the expectation term involving the advantage function:

$$\begin{aligned} \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[\sum_{t=0}^{\infty} \gamma^t A_{\pi_{\text{mar}}}(s_t, \mathbf{a}_t) \right] &= \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[\sum_{t=0}^{\infty} \gamma^t [Q_{\pi_{\text{mar}}}(s_t, \mathbf{a}_t) - V_{\pi_{\text{mar}}}(s_t)] \right] \\ &= \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[\sum_{t=0}^{\infty} \gamma^t [r_t + \gamma V_{\pi_{\text{mar}}}(s_{t+1}) - V_{\pi_{\text{mar}}}(s_t)] \right] \\ &= \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[-V_{\pi_{\text{mar}}}(s_0) + \sum_{t=0}^{\infty} \gamma^t r_t \right]. \end{aligned} \quad (23)$$

By the definition of (1) and (2), we can obtain that:

$$J(\pi_{\text{mar}}) = \mathbb{E}_{s_0, M_0, \mathbf{a}_0, s_1, M_1, \mathbf{a}_1, \dots} \left(\sum_{t=0}^{\infty} \gamma^t r_t \right), \quad (24)$$

$$V_{\pi_{\text{mar}}}(s_0) = \mathbb{E}_{M_0, \mathbf{a}_0, s_1, M_1, \mathbf{a}_1, \dots} \left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s \right], \quad (25)$$

$$J(\pi_{\text{mar}}) = \mathbb{E}_{s_0} [V_{\pi_{\text{mar}}}(s_0)]. \quad (26)$$

Therefore, (23) can be further simplified into:

$$\begin{aligned} \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[\sum_{t=0}^{\infty} \gamma^t A_{\pi_{\text{mar}}}(s_t, \mathbf{a}_t) \right] &= \mathbb{E}_{\tau \sim \bar{\pi}_{\text{mar}}} \left[-V_{\pi_{\text{mar}}}(s_0) + \sum_{t=0}^{\infty} \gamma^t r_t \right] \\ &= -J(\pi_{\text{mar}}) + J(\bar{\pi}_{\text{mar}}). \end{aligned}$$

□

A.2 Proof of Theorem 1

Theorem 1. Under the proposed conductor-based framework, a significant policy improvement inequality holds for the joint policy π_{mar} :

$$\begin{aligned} J(\bar{\pi}_{\text{mar}}) &\geq J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}} \left[\mathbb{E}_{M \sim \bar{w}(\cdot|s)} A_{\pi_{\text{mar}}}(M|s) - \text{CD}_{\text{KL}}^{\max}(w, \bar{w}) \right. \\ &\quad \left. + \mathbb{E}_{M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M) - \max C \sum_{j=1}^K w(M^j|s) \text{D}_{\text{KL}}(\pi(\cdot|s, M^j), \bar{\pi}(\cdot|s, M^j)) \right], \end{aligned} \quad (27)$$

where $C = \frac{4\gamma \max_{s,a} |A_{\pi_{\text{mar}}}(s, a)|}{(1-\gamma)^2}$, $\text{D}_{\text{KL}}^{\max}(w_k, \bar{w}) = \max_s \text{D}_{\text{KL}}(w_k(\cdot|s), \bar{w}(\cdot|s))$.

Proof. First, from the alternative objective function (14), we can obtain that:

$$\begin{aligned} L_{\pi_{\text{mar}}}(\bar{\pi}_{\text{mar}}) &= J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}, M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} [A_{\pi_{\text{mar}}}(s, \mathbf{a})] \\ &= J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}, M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} [A_{\pi_{\text{mar}}}(M|s) + A_{\pi_{\text{mar}}}(\mathbf{a}|s, M)] \\ &= J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}} [\mathbb{E}_{M \sim \bar{w}(\cdot|s)} A_{\pi_{\text{mar}}}(M|s) + \mathbb{E}_{M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M)]. \end{aligned} \quad (28)$$

Therefore, combining Theorem 1 in Kuba et al. (2022), we can derive the following inequality:

$$\begin{aligned} J(\bar{\pi}_{\text{mar}}) &\geq L_{\pi_{\text{mar}}}(\bar{\pi}_{\text{mar}}) - \text{CD}_{\text{KL}}^{\max}(\pi_{\text{mar}}, \bar{\pi}_{\text{mar}}) \\ &= J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}} \left[\mathbb{E}_{M \sim \bar{w}(\cdot|s)} A_{\pi_{\text{mar}}}(M|s) + \mathbb{E}_{M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M) \right] - \text{CD}_{\text{KL}}^{\max}(\pi_{\text{mar}}, \bar{\pi}_{\text{mar}}), \end{aligned} \quad (29)$$

where $C = \frac{4\gamma \max_{s,a} |A_{\pi_{\text{mar}}}(s, a)|}{(1-\gamma)^2}$ Kuba et al. (2022). Next, we simplify $\text{D}_{\text{KL}}^{\max}(\pi_{\text{mar}}, \bar{\pi}_{\text{mar}})$ by expressing it in terms of the instruction preference distribution $w(\cdot|s)$ and the instruction-conditional joint policy $\pi(\mathbf{a}|s, M)$:

$$\text{D}_{\text{KL}}(\pi_{\text{mar}}(\cdot|s), \bar{\pi}_{\text{mar}}(\cdot|s)) \leq \text{D}_{\text{KL}}(w(\cdot|s), \bar{w}(\cdot|s)) + \sum_{j=1}^K w(M^j|s) \text{D}_{\text{KL}}(\pi(\cdot|s, M^j), \bar{\pi}(\cdot|s, M^j)), \quad (30)$$

where the detail derivation of (30) can be found in section A.3. By taking maximum over state s , we have:

$$\text{D}_{\text{KL}}^{\max}(\pi_{\text{mar}}, \bar{\pi}_{\text{mar}}) \leq \text{D}_{\text{KL}}^{\max}(w, \bar{w}) + \max \sum_{j=1}^K w(M^j|s) \text{D}_{\text{KL}}(\pi(\cdot|s, M^j), \bar{\pi}(\cdot|s, M^j)). \quad (31)$$

In summary, (29) can be further decomposed into:

$$\begin{aligned} J(\bar{\pi}_{\text{mar}}) &\geq J(\pi_{\text{mar}}) + \mathbb{E}_{s \sim \rho_{\pi_{\text{mar}}}} \left[\mathbb{E}_{M \sim \bar{w}(\cdot|s)} A_{\pi_{\text{mar}}}(M|s) - \text{CD}_{\text{KL}}^{\max}(w, \bar{w}) \right. \\ &\quad \left. + \mathbb{E}_{M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M) - \max C \sum_{j=1}^K w(M^j|s) \text{D}_{\text{KL}}(\pi(\cdot|s, M^j), \bar{\pi}(\cdot|s, M^j)) \right]. \end{aligned}$$

□

A.3 Derivation of the Equation (30) for KL-divergence

$$\begin{aligned}
D_{\text{KL}}(\pi_{\text{mar}}(\cdot|s), \bar{\pi}_{\text{mar}}(\cdot|s)) &= \mathbb{E}_{\mathbf{a} \sim \pi_{\text{mar}}} [\log \pi_{\text{mar}}(\mathbf{a}|s) - \log \bar{\pi}_{\text{mar}}(\mathbf{a}|s)] \\
&= \sum_{\mathbf{a}} \pi_{\text{mar}}(\mathbf{a}|s) \log \frac{\pi_{\text{mar}}(\mathbf{a}|s)}{\bar{\pi}_{\text{mar}}(\mathbf{a}|s)} \\
&= \sum_{\mathbf{a}} \pi_{\text{mar}}(\mathbf{a}|s) \log \frac{\sum_{j=1}^K w(M^j|s) \pi(\mathbf{a}|s, M^j)}{\sum_{j=1}^K \bar{w}(M^j|s) \bar{\pi}(\mathbf{a}|s, M^j)} \\
&= \sum_{\mathbf{a}} \sum_{j=1}^K w(M^j|s) \pi(\mathbf{a}|s, M^j) \log \frac{\sum_{j=1}^K w(M^j|s) \pi(\mathbf{a}|s, M^j)}{\sum_{j=1}^K \bar{w}(M^j|s) \bar{\pi}(\mathbf{a}|s, M^j)} \\
&\leqslant \sum_{\mathbf{a}} \sum_{j=1}^K \left[w(M^j|s) \pi(\mathbf{a}|s, M^j) \log \frac{w(M^j|s) \pi(\mathbf{a}|s, M^j)}{\bar{w}(M^j|s) \bar{\pi}(\mathbf{a}|s, M^j)} \right] (\text{by log sum inequality}) \\
&= \sum_{\mathbf{a}} \sum_{j=1}^K w(M^j|s) \pi(\mathbf{a}|s, M^j) [\log \frac{w(M^j|s)}{\bar{w}(M^j|s)} + \log \frac{\pi(\mathbf{a}|s, M^j)}{\bar{\pi}(\mathbf{a}|s, M^j)}] \\
&= \sum_{j=1}^K w(M^j|s) \log \frac{w(M^j|s)}{\bar{w}(M^j|s)} \sum_{\mathbf{a}} \pi(\mathbf{a}|s, M^j) + \sum_{\mathbf{a}} \sum_{j=1}^K w(M^j|s) \pi(\mathbf{a}|s, M^j) \log \frac{\pi(\mathbf{a}|s, M^j)}{\bar{\pi}(\mathbf{a}|s, M^j)} \\
&= D_{\text{KL}}(w(\cdot|s), \bar{w}(\cdot|s)) + \sum_{j=1}^K w(M^j|s) D_{\text{KL}}(\pi(\cdot|s, M^j), \bar{\pi}(\cdot|s, M^j)). \tag{32}
\end{aligned}$$

A.4 Proof of Guaranteed Monotonic Joint Policy Optimization for HCPO

Before proof of monotonic improvement guarantee, we simplify certain inequality (27) to prepare for the subsequent steps. We begin by dealing with $\mathbb{E}_{M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M)$, which is vital for assessing the influence of policy updates on the system's overall performance.

$$\begin{aligned}
\mathbb{E}_{M \sim \bar{w}(\cdot|s), \mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M) &= \mathbb{E}_{M \sim \bar{w}(\cdot|s)} [\mathbb{E}_{\mathbf{a} \sim \bar{\pi}(\cdot|s, M)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M)] \\
&= \sum_{j=1}^K \bar{w}(M^j|s) [\mathbb{E}_{\mathbf{a} \sim \bar{\pi}(\cdot|s, M^j)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M^j)]. \tag{33}
\end{aligned}$$

To facilitate subsequent representations, define:

$$L_{\pi_{\text{mar}}}^{i_{1:l}}(\bar{\pi}^{i_{1:l-1}}, \hat{\pi}^{i_l}|s, M) \triangleq \mathbb{E}_{\mathbf{a}^{i_{1:l-1}} \sim \bar{\pi}^{i_{1:l-1}}(\cdot|s, M), a^{i_l} \sim \hat{\pi}^{i_l}(\cdot|s, M)} [A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M)]. \tag{34}$$

It is easy to obtain that for any $\bar{\pi}^{i_{1:l-1}}$, there is:

$$\begin{aligned}
L_{\pi_{\text{mar}}}^{i_{1:l}}(\bar{\pi}^{i_{1:l-1}}, \pi^{i_l}|s, M) &= \mathbb{E}_{\mathbf{a}^{i_{1:l-1}} \sim \bar{\pi}^{i_{1:l-1}}(\cdot|s, M), a^{i_l} \sim \pi^{i_l}(\cdot|s, M)} [A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M)] \\
&= \mathbb{E}_{\mathbf{a}^{i_{1:l-1}} \sim \bar{\pi}^{i_{1:l-1}}(\cdot|s, M)} [\mathbb{E}_{a^{i_l} \sim \pi^{i_l}(\cdot|s, M)} [A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M)]] \\
&= 0. \tag{35}
\end{aligned}$$

Consider agents updated in order $i_{1:N}$, $\forall M^j, j \in \{1, 2, \dots, K\}$, we have:

$$\begin{aligned}
\mathbb{E}_{\mathbf{a} \sim \bar{\pi}(\cdot|s, M^j)} A_{\pi_{\text{mar}}}(\mathbf{a}|s, M^j) &= \mathbb{E}_{\mathbf{a}^{i_{1:N}} \sim \bar{\pi}^{i_{1:N}}(\cdot|s, M^j)} A_{\pi_{\text{mar}}}^{i_{1:N}}(\mathbf{a}^{i_{1:N}}|s, M^j) \\
&\quad (\text{by (20)}) = \mathbb{E}_{\mathbf{a}^{i_{1:N}} \sim \bar{\pi}^{i_{1:N}}(\cdot|s, M^j)} \sum_{l=1}^N A_{\pi_{\text{mar}}}^{i_l}(\mathbf{a}^{i_{1:l-1}}, a^{i_l}|s, M^j) \\
&\quad (\text{by (34)}) = \sum_{l=1}^N L_{\pi_{\text{mar}}}^{i_{1:l}}(\bar{\pi}^{i_{1:l-1}}, \hat{\pi}^{i_l}|s, M^j). \tag{36}
\end{aligned}$$

In addition, for D_{KL} in the last item in (30), there is:

$$\begin{aligned}
D_{KL}(\pi(\cdot|s, M^j), \bar{\pi}(\cdot|s, M^j)) &= \mathbb{E}_{\mathbf{a} \sim \pi(\cdot|s, M^j)} [\log \pi(\mathbf{a}|s, M^j) - \log \bar{\pi}(\mathbf{a}|s, M^j)] \\
&= \mathbb{E}_{\mathbf{a} \sim \pi(\cdot|s, M^j)} \left[\log \left(\prod_{l=1}^N \pi^{i_l}(a^{i_l}|s, M^j) \right) - \log \left(\prod_{l=1}^N \bar{\pi}^{i_l}(a^{i_l}|s, M^j) \right) \right] \\
&= \mathbb{E}_{\mathbf{a} \sim \pi(\cdot|s, M^j)} \left[\sum_{l=1}^N \log \pi^{i_l}(a^{i_l}|s, M^j) - \sum_{l=1}^N \log \bar{\pi}^{i_l}(a^{i_l}|s, M^j) \right] \\
&= \sum_{l=1}^N \mathbb{E}_{a^{i_l} \sim \pi^{i_l}(\cdot|s, M^j), \mathbf{a}^{-i_l} \sim \pi^{-i_l}(\cdot|s, M^j)} [\log \pi^{i_l}(a^{i_l}|s, M^j) - \log \bar{\pi}^{i_l}(a^{i_l}|s, M^j)] \\
&= \sum_{l=1}^N D_{KL}(\pi^{i_l}(\cdot|s, M^j), \bar{\pi}^{i_l}(\cdot|s, M^j)). \tag{37}
\end{aligned}$$

Then, we can prove the monotonic improvement guarantee for the performance of the joint policy π_{mar} , under the following two-level policy update mechanisms:

(i) The conductor's policy $w(\cdot|s) = w_k(\cdot|s)$ is updated first according to the rule:

$$w_{k+1} = \arg \max_{\bar{w}} \left[\mathbb{E}_{s \sim \rho_{\pi_{\text{mar}, k}}, M \sim \bar{w}} A_{\pi_{\text{mar}, k}}(M|s) - CD_{KL}^{\max}(w_k, \bar{w}) \right]. \tag{38}$$

(ii) For each M^j where $j \in \{1, 2, \dots, K\}$, the agents update their policies sequentially according to the order $i_{1:N}$, with the following update rule:

$$\begin{aligned}
\pi_{k+1}^{i_l}(\cdot|s, M^j) &= \arg \max_{\bar{\pi}^{i_l}(\cdot|s, M^j)} \left[w_{k+1}(M^j|s) L_{\pi_{\text{mar}, k}}^{i_{1:l}} \left(\pi_{k+1}^{i_{1:l-1}}, \bar{\pi}^{i_l}|s, M^j \right) \right. \\
&\quad \left. - \max_s C w_k(M^j|s) D_{KL}(\pi_k^{i_l}(\cdot|s, M^j), \bar{\pi}^{i_l}(\cdot|s, M^j)) \right]. \tag{39}
\end{aligned}$$

Proof. Starting from (27), combining (38) and (39), we can obtain:

$$\begin{aligned}
J(\boldsymbol{\pi}_{\text{mar},k+1}) &\geq J(\boldsymbol{\pi}_{\text{mar},k}) + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} [\mathbb{E}_{M \sim w_{k+1}(\cdot|s)} A_{\boldsymbol{\pi}_{\text{mar},k}}(M|s) - \text{CD}_{\text{KL}}^{\max}(w_k, w_{k+1})] \\
&\quad + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} \left[\mathbb{E}_{M \sim w_{k+1}(\cdot|s), \mathbf{a} \sim \boldsymbol{\pi}_{k+1}(\cdot|s, M)} A_{\boldsymbol{\pi}_{\text{mar},k}}(\mathbf{a}|s, M) \right. \\
&\quad \left. - \max_s C \sum_{j=1}^K w_k(M^j|s) \text{D}_{\text{KL}}(\boldsymbol{\pi}_k(\cdot|s, M^j), \boldsymbol{\pi}_{k+1}(\cdot|s, M^j)) \right] \\
&= J(\boldsymbol{\pi}_{\text{mar},k}) + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} [\mathbb{E}_{M \sim w_{k+1}(\cdot|s)} A_{\boldsymbol{\pi}_{\text{mar},k}}(M|s) - \text{CD}_{\text{KL}}^{\max}(w_k, w_{k+1})] \\
&\quad + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} \left[\sum_{j=1}^K w_{k+1}(M^j|s) [\mathbb{E}_{\mathbf{a} \sim \boldsymbol{\pi}_{k+1}(\cdot|s, M^j)} A_{\boldsymbol{\pi}_{\text{mar},k}}(\mathbf{a}|s, M^j)] \right. \\
&\quad \left. - \max_s C \sum_{j=1}^K w_k(M^j|s) \text{D}_{\text{KL}}(\boldsymbol{\pi}_k(\cdot|s, M^j), \boldsymbol{\pi}_{k+1}(\cdot|s, M^j)) \right] \text{ (combine (33))} \\
&= J(\boldsymbol{\pi}_{\text{mar},k}) + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} [\mathbb{E}_{M \sim w_{k+1}(\cdot|s)} A_{\boldsymbol{\pi}_{\text{mar},k}}(M|s) - \text{CD}_{\text{KL}}^{\max}(w_k, w_{k+1})] \\
&\quad + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} \left[\sum_{j=1}^K w_{k+1}(M^j|s) \sum_{l=1}^N L_{\boldsymbol{\pi}_{\text{mar},k}}^{i_{1:l}}(\boldsymbol{\pi}_{k+1}^{i_{1:l-1}}, \pi_{k+1}^{i_l}|s, M^j) \right. \\
&\quad \left. - \max_s C \sum_{j=1}^K w_k(M^j|s) \sum_{l=1}^N \text{D}_{\text{KL}}(\pi_k^{i_l}(\cdot|s, M^j), \pi_{k+1}^{i_l}(\cdot|s, M^j)) \right] \text{ (combine (36)(37))} \quad (40) \\
&\geq J(\boldsymbol{\pi}_{\text{mar},k}) + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} [\mathbb{E}_{M \sim w_{k+1}(\cdot|s)} A_{\boldsymbol{\pi}_{\text{mar},k}}(M|s) - \text{CD}_{\text{KL}}^{\max}(w_k, w_{k+1})] \\
&\quad + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} \left[\sum_{l=1}^N \sum_{j=1}^K [w_{k+1}(M^j|s) L_{\boldsymbol{\pi}_{\text{mar},k}}^{i_{1:l}}(\boldsymbol{\pi}_{k+1}^{i_{1:l-1}}, \pi_{k+1}^{i_l}|s, M^j) \right. \\
&\quad \left. - \max_s C w_k(M^j|s) \text{D}_{\text{KL}}(\pi_k^{i_l}(\cdot|s, M^j), \pi_{k+1}^{i_l}(\cdot|s, M^j))] \right] \\
&\geq J(\boldsymbol{\pi}_{\text{mar},k}) + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} [\mathbb{E}_{M \sim w_k(\cdot|s)} A_{\boldsymbol{\pi}_{\text{mar},k}}(M|s) - \text{CD}_{\text{KL}}^{\max}(w_k, w_k)] \\
&\quad + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} \left[\sum_{l=1}^N \sum_{j=1}^K [w_{k+1}(M^j|s) L_{\boldsymbol{\pi}_{\text{mar},k}}^{i_{1:l}}(\boldsymbol{\pi}_{k+1}^{i_{1:l-1}}, \pi_k^{i_l}|s, M^j) \right. \\
&\quad \left. - \max_s C w_k(M^j|s) \text{D}_{\text{KL}}(\pi_k^{i_l}(\cdot|s, M^j), \pi_{k+1}^{i_l}(\cdot|s, M^j))] \right] \text{ (combine (38)(39))} \\
&= J(\boldsymbol{\pi}_{\text{mar},k}) + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} [\mathbb{E}_{M \sim w_k(\cdot|s)} A_{\boldsymbol{\pi}_{\text{mar},k}}(M|s) - 0] \\
&\quad + \mathbb{E}_{s \sim \rho_{\boldsymbol{\pi}_{\text{mar},k}}} \left[\sum_{l=1}^N \sum_{j=1}^K [w_{k+1}(M^j|s) L_{\boldsymbol{\pi}_{\text{mar},k}}^{i_{1:l}}(\boldsymbol{\pi}_{k+1}^{i_{1:l-1}}, \pi_k^{i_l}|s, M^j) - 0] \right].
\end{aligned}$$

By the meaning of the advantage function and the equation (35), there is:

$$\mathbb{E}_{M \sim w_k(\cdot|s)} A_{\boldsymbol{\pi}_{\text{mar},k}}(M|s) = 0, \quad L_{\boldsymbol{\pi}_{\text{mar},k}}^{i_{1:l}}(\boldsymbol{\pi}_{k+1}^{i_{1:l-1}}, \pi_k^{i_l}|s, M^j) = 0.$$

Therefore, (40) can be reduced to:

$$J(\boldsymbol{\pi}_{\text{mar},k+1}) \geq J(\boldsymbol{\pi}_{\text{mar},k}) + 0 = J(\boldsymbol{\pi}_{\text{mar},k}). \quad (41)$$

□

B. Algorithms

Algorithm 1: Pseudocode of HCPO

Input: Minibatch size B_1 , number of agents N , training times Λ , steps per episode T

- 1: **Initialization:** Global V-value network $\{\phi_0\}$, centralized conductor's actor network $\{\Psi_0\}$, local conductors' actor networks $\{\psi_0^i, \forall i \in \mathcal{N}\}$, agents' actor networks $\{\theta_0^i, \forall i \in \mathcal{N}\}$, replay buffer \mathcal{B}_1
 - 2: **for** $k = 0, 1, \dots, \Lambda - 1$ **do**
 - 3: Collect a set of trajectories by running $\pi_{\text{mar},k}(\mathbf{a}_t|s_t) = \mathbb{E}_{M_t \sim w^{\Psi_k(\cdot|s_t)}} \pi_{\theta_k}(\mathbf{a}_t|s_t, M_t)$ and save transitions $\{(s_t, M_t, \mathbf{a}_t, s_{t+1}, r_t), \forall t \in T\}$ into \mathcal{B}_1
 - 4: Sample a random minibatch of B_1 episodes from replay buffer \mathcal{B}_1
 - 5: Compute $A_{\pi_{\text{mar},k}}(M_{b,t}|s_{b,t})$ and $A_{\pi_{\text{mar},k}}(\mathbf{a}_{b,t}|s_{b,t}, M_{b,t})$ based on global V-value network with generalized advantage estimation (GAE)
 - 6: Update $\Psi_{k+1} = \Psi_k + \alpha^{j_1} \hat{\beta}_{\text{conductor},k} \hat{x}_{\text{conductor},k}$ as designed in Algorithm 2
 - 7: Draw a random permutation of agents $i_{1:N}$
 - 8: Set $\Theta^{i_1}(\mathbf{a}_{b,t}|s_{b,t}, M_{b,t}) = A_{\pi_{\text{mar},k}}(\mathbf{a}_{b,t}|s_{b,t}, M_{b,t})$
 - 9: **for** agent $i_l = i_1, \dots, i_N$ **do**
 - 10: Update $\psi_{k+1}^{i_l}$ by minimizing the cross-entropy between $\psi_k^{i_l}$ and Ψ_{k+1} : $\mathcal{L}_{ce}^{i_l} = -\mathbb{E}_{w^{\Psi_{k+1}}} [\log \psi_k^{i_l}]$
 - 11: Update agent i_l 's policy by $\theta_{k+1}^{i_l} = \theta_k^{i_l} + \alpha^{j_2} \hat{\beta}_{\text{agent},k}^{i_l} \hat{x}_{\text{agent},k}^{i_l}$ as designed in Algorithm 3
 - 12: Compute
$$\Theta^{i_{1:l+1}}(\mathbf{a}_{b,t}|s_{b,t}, M_{b,t}) = \frac{\pi_{\theta_{k+1}^{i_l}}^{i_l} \left(a_{b,t}^{i_l} | o_{b,t}^{i_l}, M_{b,t} \right)}{\pi_{\theta_k^{i_l}}^{i_l} \left(a_{b,t}^{i_l} | o_{b,t}^{i_l}, M_{b,t} \right)} \Theta^{i_{1:l}}(\mathbf{a}_{b,t}|s_{b,t}, M_{b,t})$$
 unless $l = N$
 - 13: **end for**
 - 14: Update V-value network by following formula:
$$\phi_{k+1} = \arg \min_{\phi} \frac{1}{B_1 T} \sum_{b=1}^{B_1} \sum_{t=0}^T \left(V_{\phi}(s_t) - \hat{R}_t \right)^2$$
 - 15: Clear the replay buffer \mathcal{B}_1 and minibatch B_1
 - 16: **end for**
-

Algorithm 2: Conjugate Gradient Approach for Updating Centralized Conductor's Policy Parameters

Input: Minibatch size B_1 , episode number k , steps per episode T

Output: New centralized conductor's policy parameter Ψ_{k+1}

- 1: Estimate the gradient of the conductor's maximization objective

$$\hat{g}_{conductor,k} = \frac{1}{B_1} \sum_{b=1}^{B_1} \sum_{t=1}^T \nabla_{\Psi_k} \log w^{\Psi_k}(M_{b,t}|s_{b,t}) A_{\pi_{\text{mar},k}}(M_{b,t}|s_{b,t})$$

Use the conjugate gradient algorithm to compute the update direction

$$\hat{x}_{conductor,k} \approx \hat{\mathbf{H}}_{conductor,k}^{-1} \hat{g}_{conductor,k}$$

where $\hat{\mathbf{H}}_{conductor,k}$ is the Hessian of the average KL-divergence

$$\frac{1}{B_1 T} \sum_{b=1}^{B_1} \sum_{t=1}^T D_{\text{KL}}(w^{\Psi_k}(\cdot|s_{b,t}), w^{\Psi}(\cdot|s_{b,t}))$$

- 2: Estimate the maximal step size allowing for meeting the KL-constraint

$$\hat{\beta}_{conductor,k} \approx \sqrt{\frac{2\delta_1}{(\hat{x}_{conductor,k})^\top \hat{\mathbf{H}}_{conductor,k} \hat{x}_{conductor,k}}}$$

- 3: Update centralized conductor's policy by $\Psi_{k+1} = \Psi_k + \alpha^{j_1} \hat{\beta}_{conductor,k} \hat{x}_{conductor,k}$
-

Algorithm 3: Conjugate Gradient Approach for Updating Local Agent's Policy Parameters

Input: Minibatch size B_1 , episode number k , steps per episode T , agent index i_l

Output: New Agent's policy parameter $\theta_{k+1}^{i_l}$

- 1: Estimate the gradient of the agent's maximization objective

$$\hat{g}_{agent,k}^{i_l} = \frac{1}{B_1} \sum_{b=1}^{B_1} \sum_{t=1}^T w^{\Psi_{k+1}}(M_{b,t}|s_{b,t}) \nabla_{\theta_k^{i_l}} \log \pi_{\theta_k^{i_l}}^{i_l}(a_{b,t}^{i_l} | o_{b,t}^{i_l}, M_{b,t}) \Theta^{i_{1:l}}(a_{b,t}|s_{b,t}, M_{b,t})$$

Use the conjugate gradient algorithm to compute the update direction

$$\hat{x}_{agent,k}^{i_l} \approx \hat{\mathbf{H}}_{agent,k}^{i_l} \hat{g}_{agent,k}^{i_l}$$

where $\hat{\mathbf{H}}_{agent,k}^{i_l}$ is the Hessian of the average KL-divergence

$$\frac{1}{B_1 T} \sum_{b=1}^{B_1} \sum_{t=1}^T w^{\Psi_k}(M_{b,t}|s_{b,t}) D_{\text{KL}}\left(\pi_{\theta_k^{i_l}}^{i_l}(\cdot|o_{b,t}^{i_l}, M_{b,t}), \pi_{\theta_k^{i_l}}^{i_l}(\cdot|o_{b,t}^{i_l}, M_{b,t})\right)$$

- 2: Estimate the maximal step size allowing for meeting the KL-constraint

$$\hat{\beta}_{agent,k}^{i_l} \approx \sqrt{\frac{2\delta_2}{(\hat{x}_{agent,k}^{i_l})^\top \hat{\mathbf{H}}_{agent,k}^{i_l} \hat{x}_{agent,k}^{i_l}}}$$

- 3: Update agent i_l 's policy by $\theta_{k+1}^{i_l} = \theta_k^{i_l} + \alpha^{j_2} \hat{\beta}_{agent,k}^{i_l} \hat{x}_{agent,k}^{i_l}$
-

C. Experimental Details

C.1 StarCraftII Multi-agent Challenge

In the SMAC environment, we basically adhere to the official implements and hyperparameter settings of MAVEN¹ Mahajan et al. (2019), HAPPO and HATRPO² Kuba et al. (2022), A2PO³ Wang et al. (2023), HAA2C⁴ Zhong et al. (2024), as shown in Table 1. We adopt the evaluation method from MAPPO Yu et al. (2022) and compare HCPO against other algorithms on five maps. After each training iteration, 32 evaluation games are played and the winning rate of these 32 games is calculated. Finally, we take the median winning rate of the last ten evaluations as the performance metric for each random seed and report the average median winning rate based on the five random seeds in Table 2. We can observe that HCPO achieves the highest average median winning rate across five maps, reaching an impressive 97.82%. This result highlights its superior performance compared to other algorithms. Moreover, HCPO also exhibits the lowest standard deviation, demonstrating its high stability.

hyperparameters	value	hyperparameters	value	hyperparameters	value
critic lr	5e-4	optimizer	Adam	stacked-frames	1
gamma	0.95	gamma in corridor	0.99	optim eps	1e-5
batch size	3200	gain	0.01	hidden layer	1
training threads	32	actor network	mlp	num mini-batch	1
rollout threads	16	hypernet embed	64	max grad norm	10
episode length	200	activation	ReLU	hidden layer dim	64
use huber loss	True	conductor kl-threshold	0.01	kl-threshold	0.06
accept-ratio	0.5	K	10		

Table 1: Common hyperparameters in SMAC

Task	Difficulty	HCPO	HAPPO	HATRPO	HAA2C	A2PO	MAVEN
3s5z	hard	100(0.0)	89.1(2.3)	100(1.4)	67.2(11.8)	98.4(1.3)	0.0(0.0)
5m_vs_6m	hard	93.8(2.9)	90.6(5.4)	70.3(8.7)	0.0(0.0)	92.2(5.9)	43.8(1.4)
8m_vs_9m	hard	100(0.7)	81.2(6.9)	90.6(2.4)	18.8(2.7)	93.8(4.5)	18.8(2.9)
10m_vs_11m	hard	98.4(1.6)	85.9(8.4)	93.8(2.9)	20.3(11.1)	96.9(2.2)	18.8(2.6)
corridor	super	96.9(1.6)	96.9(0.7)	90.6(2.3)	85.9(5.7)	93.8(2.4)	40.6(4.1)
Overall	/	97.82(1.36)	88.74(4.74)	89.06(3.54)	38.44(6.26)	95.02(3.26)	24.4(2.2)

Table 2: Average evaluation median winning rate and standard deviation (across five seeds) within SMAC scenarios for distinct methods

Taking the *3s5z* map as an example, after the completion of the training process (10 million steps) for all agents' local conductors and local actors, we visualize the gameplay scenarios involving our allies and the enemies in Figure 9. Figure 9a depicts the initial phase of the game, with both teams launching their attacks. The Figure 9b captures a moment where one of the stalkers is intentionally drawing the enemy fire, acting as a decoy to distract and absorb the enemy's attention and attacks. This instruction allows the other two stalkers to approach from behind the zealots, positioning them for a surprise attack on the enemy's flank. The green circles highlight the units under our control, with the health and energy bars visible above each unit, indicating their current status. The blue line pointing towards the enemy unit signifies the attack direction of our stalker, demonstrating the coordination between our units. Figure 9c illustrates a strategic (instruction-driven) moment where our forces are effectively split into two groups. The green arrows represent the movement direction of our stalkers. These two stalkers have identified allies in the yellow circle on the right with low health and automatically move to provide support. This self-directed action, triggered by the agent's local assessment of the situation (observation and instruction), exemplifies the autonomous decision-making capabilities of our agents under the guidance of HCPO's latent instructions. These instructions underscore the significance of actions and strategic positioning in achieving tactical advantage in the game. The experiments are performed on a server with 3 RTX 4090 GPUs, and 3 Xeon(R) Gold 6430 16-core CPUs, and 128GB Ram.

¹<https://github.com/starry-sky6688/MARL-Algorithms/>

²<https://github.com/PKU-MARL/TRPO-PPO-in-MARL>

³<https://github.com/xihuai18/A2PO-ICLR2023>

⁴<https://github.com/PKU-MARL/HARL>

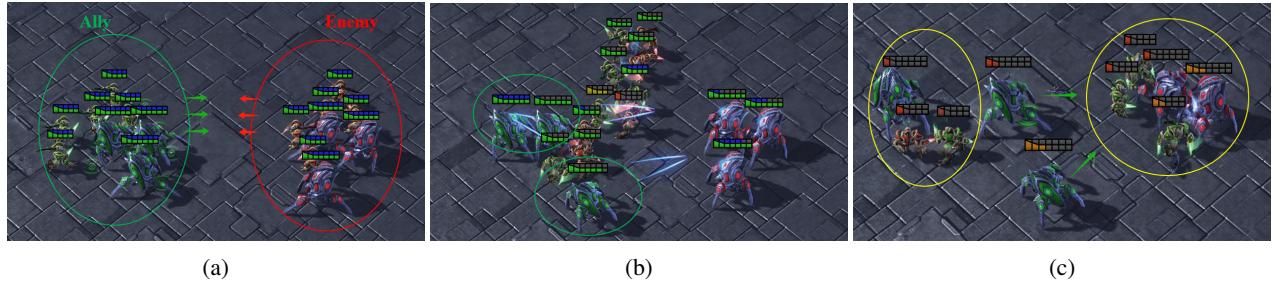


Figure 9: Effective coordination in SMAC on the $3s5z$ map: A visual analysis of agent strategies.

C.2 Multi-agent MuJoCo

The detailed experimental hyperparameters are listed in Table 3. We present a comparison of the return performance of different baselines across four tasks in Figure 10. The median return values of HCPO algorithm reach the highest levels, demonstrating its superior performance in these tasks. Additionally, the return values exhibit a narrow interquartile range, indicating high stability across multiple experimental runs. Less outliers further imply the robustness of HCPO during training, as it generates returns without being compromised by occasional suboptimal runs. In summary, these findings suggest that HCPO maintains efficiency and reliability across a variety of tasks.

In Figure 11, we employ the t-SNE (t-Distributed Stochastic Neighbor Embedding) technique to visualize the states explored by the HCPO, HATRPO, and A2PO during the early stages of training in the *Walker2d-v2-6x1*. Each dot in the plots represents a state explored by the algorithms. We use orange shading to indicate state probability density within each grid cell, defined as the fraction of samples falling in that cell relative to the total number of samples, with darker colors representing higher visitation frequencies. By examining the color distribution across the plots, we can visually assess the areas and densities explored by each algorithm. The figure also provides the average nearest neighbor distance (Avg. Dist.) and entropy values for each algorithm. A larger average nearest neighbor distance (e.g., HCPO’s 0.72) implies greater separation among visited states, reflecting better coverage diversity, while entropy measures exploration breadth. In summary, these metrics show that HCPO exhibits well-balanced exploration characteristics.

hyperparameters	value	hyperparameters	value	hyperparameters	value
critic lr	5e-3	optimizer	Adam	num mini-batch	1
gamma	0.99	optim eps	1e-5	batch size	4000
gain	0.01	hidden layer	1	training threads	8
std y coef	0.5	actor network	mlp	rollout threads	20
std x coef	1	max grad norm	10	episode length	200
activation	ReLU	hidden layer dim	128	eval episode	40
conductor kl-threshold	0.01	kl-threshold	0.005	accept-ratio	0.5
K	10				

Table 3: Common hyperparameters in MA-MuJoCo

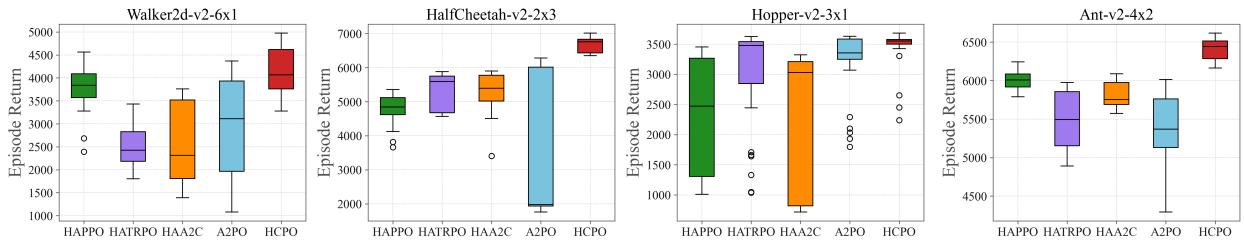


Figure 10: Comparison of the return performance of different baselines in the MuJoCo task.

C.3 Multi-agent Particle Environment

We present our detailed experimental hyperparameters in Table 4. The results of each algorithm under three random seeds on MPE benchmark are shown in Figure 12. In general, HCPO demonstrates comparable performance across all three tasks,

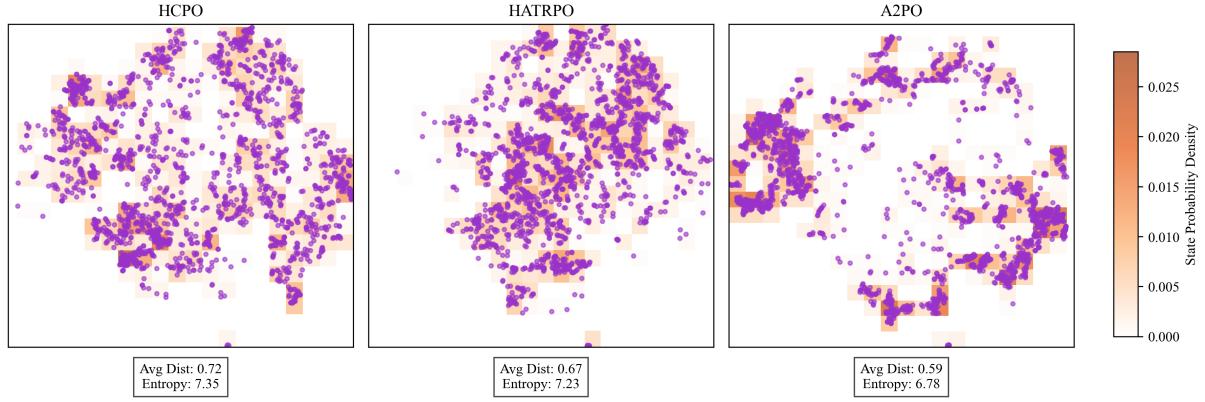


Figure 11: Exploration comparison: t-SNE visualization and entropy analysis in *Walker2d-v2-6*.

highlighting its adaptability and effectiveness in multi-agent settings. Specifically, in *simple_spread_v2-discrete* and *simple_speaker_listener_v3-discrete* tasks, HCPO exhibits rapid policy improvement in the early stage of training (0-2 million steps). This indicates that HCPO algorithm has a high cooperative efficiency and sufficient exploration. Although HCPO shows slower initial cooperation efficiency in the *simple_reference_v2-discrete* task compared to A2PO, it ultimately outperforms A2PO in final convergence performance, confirming its long-term learning advantage. Furthermore, compared with HATRPO and A2PO, HCPO algorithm shows significant stability and robustness. In summary, HCPO not only maintains competitive convergence speed but also matches or exceeds the performance of the strong baselines through its hierarchical mechanism.

hyperparameters	value	hyperparameters	value	hyperparameters	value
actor lr	5e-4	critic lr	5e-4	actor mini batch	1
critic mini batch	1	gamma	0.99	batch size	4000
network	mlp	linear lr decay	False	critic epoch	5
clip param	0.2	entropy coef	0.01	backtrack coef	0.8
conductor kl-threshold	0.01	kl-threshold	0.005	accept-ratio	0.5
K	10	eval episode	40		

Table 4: Common hyperparameters in MPE

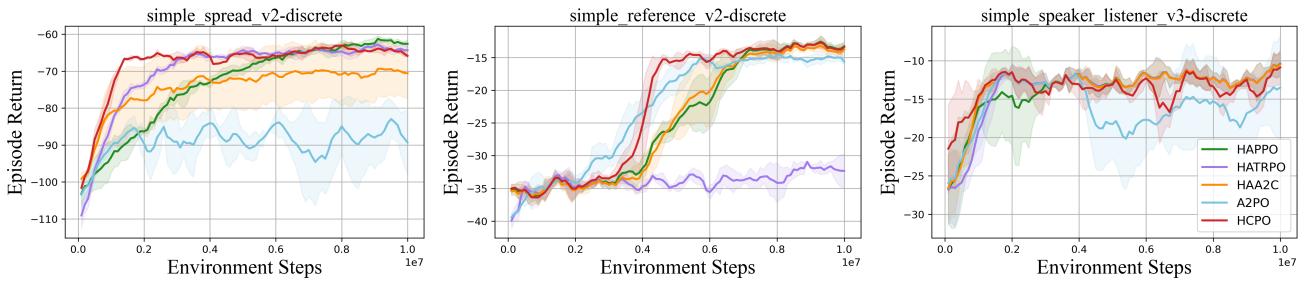


Figure 12: Performance comparison of HCPO and other strong MARL algorithms across different MPE tasks.

C.4 Ablation Studies

For several key components of HCPO, we conduct ablation studies with results shown in Figure 13. In Figure 13a, we explore the impact of removing the conductor and varying the number of instructions (hyperparameter K) on performance. HCPO with the conductor shows a faster increase in winning rate and a higher final rate than without a conductor, demonstrating its effectiveness in boosting cooperation efficiency. The performance is also influenced by K . Although increasing K enhances adaptability to the environment, beyond a certain threshold, further increments in K introduce unnecessary complexity without substantial performance gains. Therefore, it is important to balance performance and resource consumption when selecting K .

In Figure 13b, we examine the hyperparameter δ_1 , which represents the KL-divergence constraint of centralized conductor's policy. Results show that HCPO is relatively insensitive to δ_1 , but the configuration with $\delta_1 = 0.01$ demonstrates better performance compared to the other settings. In Figure 13c, we evaluate HCPO under four conductor configurations: a centralized conductor with global information, a random conductor (non-learning baseline), no conductor and local conductors only based on local observations (trained via cross-entropy learning, constituting the core of our HCPO algorithm). Figure 13d presents the final episode returns in a boxplot format. The results show that HCPO with local conductors achieves a median return comparable to that of a centralized conductor, while substantially outperforming the variant without any conductor. Furthermore, replacing the learned instruction preference distribution with a non-learning conductor that outputs uniformly random instructions leads to inferior performance. These findings collectively validate the effectiveness of our proposed update mechanisms.

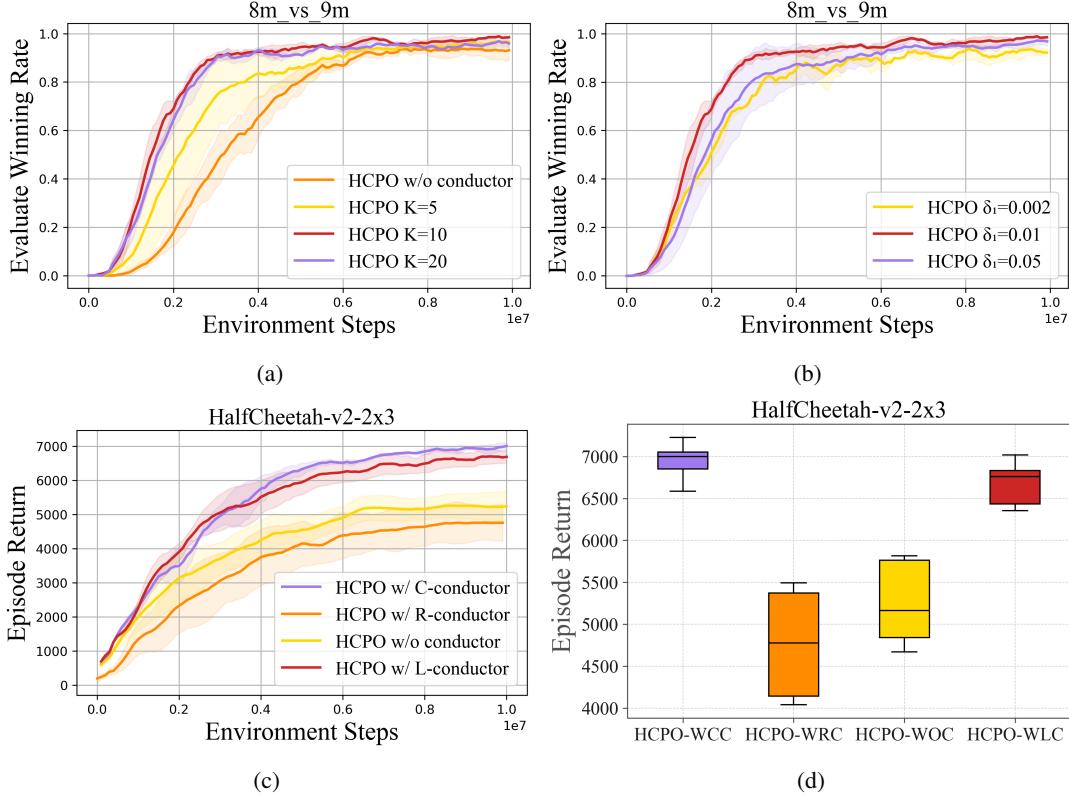


Figure 13: Ablation studies.