Decentralized Consensus Inference-Based Hierarchical Reinforcement Learning for Multiconstrained UAV Pursuit-Evasion Game

Yuming Xiang, Sizhao Li, Rongpeng Li, *Senior Member, IEEE*, Zhifeng Zhao, *Senior Member, IEEE*, and Honggang Zhang, *Fellow, IEEE*

***Abstract*—Multiple quadrotor uncrewed aerial vehicles (UAVs) systems have garnered widespread research interest and fostered tremendous interesting applications, especially in multiconstrained pursuit-evasion games (MC-PEGs). The cooperative evasion and formation coverage (CEFC) task, where the UAV swarm aims to maximize formation coverage across multiple target zones while collaboratively evading predators, belongs to one of the most challenging issues in MC-PEGs, especially under communication-limited constraints. This multifaceted problem, which intertwines responses to obstacles, adversaries, target zones, and formation dynamics, brings up significant high-dimensional complications in locating a solution. In this article, we propose a novel two-level framework [i.e., consensus inference-based hierarchical reinforcement learning (CI-HRL)], which delegates target localization to a high-level policy, while adopting a low-level policy to manage obstacle avoidance, navigation, and formation. Specifically, in the high-level policy, we develop a novel multiagent reinforcement learning (RL) module, consensus-oriented multiagent communication (ConsMAC), to enable agents to perceive global information and establish consensus from local states by effectively aggregating neighbor messages. Meanwhile, we leverage an alternative training-based MAPPO (AT-M) and policy distillation to accomplish the low-level control. The experimental results, including the high-fidelity software-in-the-loop (SITL) simulations, validate that CI-HRL provides a superior solution with enhanced swarm's collaborative evasion and task completion capabilities.**

***Index Terms*—Cooperative evasion and formation coverage (CEFC), decentralized consensus inference, hierarchical model, multiagent reinforcement learning (MARL), multiquadcopter motion planning.**

NOMENCLATURE

AT-M	Alternative training-based MAPPO.
CEFC	Cooperative evasion and formation coverage.
CI-HRL	Consensus inference-based hierarchical reinforcement learning.
ConsMAC	Consensus-oriented multiagent communication.
MC-PEG	Multiconstrained pursuit-evasion games.
SITL	Software-in-the-loop.

Received 23 May 2024; revised 25 February 2025 and 29 May 2025; accepted 21 June 2025. Date of publication 18 July 2025; date of current version 9 October 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2024YFE0200600; in part by Zhejiang Provincial Natural Science Foundation of China under Grant LR23F010005, and in part by Information Technology Center and State Key Laboratory of CAD&CG, Zhejiang University. *(Yuming Xiang and Sizhao Li contributed equally to this work.) (Corresponding author: Rongpeng Li.)*

Yuming Xiang, Sizhao Li, and Rongpeng Li are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: xiangym1999@zju.edu.cn; liszh5@zju.edu.cn; lirongpeng@zju.edu.cn).

Zhifeng Zhao is with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, and also with Zhejiang Lab, Hangzhou 311121, China (e-mail:zhaozf@zhejianglab.com).

Honggang Zhang is with the School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China (e-mail: honggang.zhang@ieee.org).

Digital Object Identifier 10.1109/TNNLS.2025.3582909

I. INTRODUCTION

NOWADAYS, quadrotor uncrewed aerial vehicles (UAVs) have demonstrated great potential in costly or human-unfriendly tasks (e.g., disaster response [1]), due to their agility, cost-effectiveness, and compact size. Nevertheless, the UAV swarm is likely to be exposed to an adversarial environment, where a hostile factor or agent might attack the affiliated members, and must respond promptly to boost the survival opportunity. Typically, such a CEFC scenario is formulated as an MC-PEG [2], wherein preys (i.e., UAVs) shall maximize the ratio of accomplishing planned missions while avoiding attacks from predators (i.e., the hostile attacker) [3]. However, as the complexity of MC-PEGs or environmental constraints escalates, conventional control algorithms face several aspects of prominent shortcomings, such as oversimplified predator strategies (e.g., fixed trajectory) [4], lack of intergroup cooperation [5], limitation to one single formation pattern [6], and unrealistic presumption on the availability of global information [7], [8].

Benefiting from the robust adaptability in complex environments, multiagent reinforcement learning (MARL)-based approaches [9], [10], [11] have been widely adopted. For instance, QMIX [12] and its variants [13] have yielded appealing results in complex multiagent scenarios, and heterogeneous-agent soft actor-critic (HASAC) [14], as the latest state-of-the-art (SOTA) algorithm, has also demonstrated outstanding capabilities in many benchmarks. While these advancements are noteworthy, recent MARL implementations in practical systems still heavily rely on multiagent proximal policy optimization (MAPPO) [15], [16] due to its stability and computational efficiency [17]. However, existing MAPPO-based multirobot control frameworks [8] typically adopt oversimplified assumptions (e.g., global obstacle coordinates and small-scale scenarios), limiting their applicability to real-world environments. Nevertheless, for complex multitasks or MC-PEGs, direct application of MAPPO often fails to achieve satisfactory convergence [18], necessitating the employment of advanced methodologies (e.g., hierarchical models [19] or alternative training [20]).

In the framework of MARL, the centralized training with decentralized execution (CTDE) architecture acts as a foundational solution [9]. Accordingly, many variants of CTDE [21] have been proposed to improve the execution perfor-

mance by devising a communication module to allow agents to explicitly or implicitly exchange their local information during the training. Contingent on a communication network or leader-follower assumption, these MARL approaches generally face some communication-performance dilemmas. For example, [13] unveils that a globally shared observation might generate a significant amount of redundant information and even yield less competitive results than the case with partial local observation only. Therefore, it becomes critical to design some consensus inference algorithms to effectively guide the interaction between agents. In this regard, conventional algorithms [22] generally formulate an optimization problem and utilize control theory to produce a solution. Nevertheless, these algorithms lack the essential flexibility and cannot be easily merged into MARL methods [23]. Meanwhile, attributed to a closed-box deep neural network (DNN), the communication module in [24] and [25] only transmits the local information in a blunt manner, which could not unleash its potential to the full extent (e.g., unable to infer and forward global information during the execution) and fails to filter the meaningful communication content, thus being less competent to handle complex scenarios. As a remedy, the opponent modeling approaches [26], [27] interpret the communication content as the speculated future actions of other agents. However, in partially observable scenarios, this approach suffers from speculation inconsistency, as individual agents possibly make different speculations according to their local observations [28]. Thus, it remains challenging to infer the consensus and perceive consistent information from diversified, limited local information. Alternatively, hierarchical reinforcement learning (HRL) is considered to effectively deal with these underlying difficulties [29]. In HRL, high-level policies are cooperatively learned to focus on subgoals (e.g., to which position), while the low-level policies are designed for completing subgoal-related basic operations (e.g., specific movements and obstacle avoidance) [19]. The coordinated interplay between two-level policies often achieves effects in complex tasks that surpass the capabilities of a single-level structure [30]. Nevertheless, integrating consensus mechanisms into HRL adds to the training difficulty, especially when agents must make decisions based on incomplete and diverse local observations.

In this article, toward the CEFC control in a partially observed environment, we propose a novel decentralized CI-HRL framework. To be specific, to tackle the global collaboration problem of agents with local observations, we incorporate a hierarchical CTDE architecture with both high and low levels. In particular, the high-level policy is designed to select appropriate anchor points (i.e., target positions), according to the state of neighbors and predators on top of ConsMAC, while the low-level policy, which is implemented by alternative training and policy distillation, is responsible for adaptive formation navigation and obstacle avoidance mandated by the high-level decision. Compared with the existing work, the contribution of our article can be summarized as follows.

1) We present a hierarchical framework for a single-pursuer-multiple-evader MC-PEG. Specifically, the high-level policy provides appropriate anchor points and determines the target selection policy, while the low-level policy resolves motion control (i.e., formation, navigation, and obstacle avoidance), enabling UAVs to navigate and adapt to dynamic and uncertain CEFC environments effectively.
2) On top of alternative training-based MAPPO (AT-M), we implement an efficient multipolicy-distilled model for low-level decentralized adaptive formation with obstacle avoidance, which is capable of adapting to agent quantity changes, reducing the training cost, and improving the obstacle avoidance and formation performance.
3) The high-level policy learns a distributed target selection and division policy, based on an interagent unified understanding of the current global state provided by ConsMAC. Notably, the high-level reinforcement learning (RL)-based module and ConsMAC are trained alternately to align consensus inference and policy making.
4) Through extensive simulations in both multiagent particle environment (MPE) and SITL environment in Gazebo, we demonstrate the effectiveness and superiority of our framework over existing models.

To improve readability, we summarize a list of abbreviations above. The remainder of the article is organized as follows. Section II briefly introduces the related works. Section III presents the system model and formulates the problem. Section IV provides the details of the proposed framework. In Section V, we introduce the experimental results and discussions. Finally, Section VI concludes the article.

II. Related Work

A. DRL for PEG-Based UAV Systems

The PEG has been extensively studied in UAV systems due to its flexible requirements, and DRL has been proved effective for environmental awareness and decision control capacity [7], [31], [32]. However, the literature above only focuses on a single, simplified evader and is contingent on an over-idealistic full connectivity assumption. For example, Young and La [4] proposed a hierarchical system integrating flocking control and RL for multiagents to evade a predefined pursuer. But it implements one invariant formation and oversimplifies the pursuer policy. Ref. [33] combines a mix-attention and independent PPO (IPPO) algorithm to enhance the agents' adaptation in the multipursuer-multievader PEG. Notably, these methods neglect the cooperation among evaders and have not considered the downstream tasks such as target covering and formation maintaining. While Deng and Kong [34] designed a collaborative pursuit-defense strategy in a firefighting task, it does not take account of the possible existence of obstacles and assumes full connectivity. Different from these existing works, we address downstream tasks for the communication-limited UAV swarm and aim to develop a decentralized policy that optimizes the completion of the CEFC task while considering communication constraints and collision avoidance.

B. Hierarchical Reinforcement Learning

HRL [29] excels in simplifying complex tasks into manageable subtasks for long-term, multistep problem-solving. Classical research in HRL aims to optimize the discovery and use of subtasks and to develop algorithms supporting hierarchical structure learning [35], [36]. Option-based methods [37] enable agents to dynamically discover subtasks through environmental exploration, merging into a hierarchical policy. However, these approaches may incur higher exploration

and training expenses, and the discovered subtasks might be incomplete or suboptimal, potentially impacting overall performance quality. In our scenario, UAVs must consider both formation obstacle avoidance and cooperative pursuit avoidance, without requiring complex skill learning. To address this, we opt for classical hierarchical policy learning, where subtasks are predefined. Here, the low-level policy is trained independently and then integrated into the training of the high-level policy. This approach avoids the mutual influence and excessive difficulty of concurrently training multiple policies [38].

C. MARL With Communication

To guarantee agents with only local information to cooperate meaningfully, CTDE is widely adopted in recent MARL methods [9]. However, the partial observability in a CTDE-based multiagent environment can undermine the coordination between agents [28]. With few exceptions like HASAC [14], most recent works introduce a communication module to effectively mitigate this challenge [21]. For example, TarMAC [25] leverages the attention mechanism in the communication policy to aggregate the messages from their neighbors. However, these models do not clearly define the content and significance of communication. Instead, they treat the communication network as a closed box, thus reducing message interpretability and compromising their effectiveness in managing complex scenarios. To address this, recent studies have incorporated opponent modeling to better interpret communication content. In intention sharing (IS) [26] and ToM2C [27], each agent is designed to predict the future actions of its teammates and utilizes these predictions as the substance of communication messages. However, in partially observable scenarios, agents possibly derive varied conjectures based on their individual perceptions, and this speculation heterogeneity could potentially disrupt the agent's decision-making process. Meanwhile, multi-agent communication via self-supervised information aggregation (MASIA) [13] and neighboring variational information flow (NVIF) [11] employ supervised learning and an autoencoder to reduce the redundancy of communication. Particularly, NVIF compresses local information without linking it to global information, whereas MASIA preserves the global state but assumes full communication. Under limited communication, merely transmitting raw observations through MASIA fails to convey sufficient information for overall movement trends. In contrast, ConsMAC helps agents combine local observations and communication messages to infer a consensus on the global state, effectively addressing these challenges.

III. Preliminaries and System Model

A. System Model

In this article, we consider a PEG scenario where a well-formed swarm of UAVs flies and attempts to reach target areas across an obstacle-cluttered space. In particular, the UAV flock is fully decentralized with a limited communication range.

1) Quadcopter Model: The acceleration vector of each quadrotor UAV is defined as $[u_x, u_y, u_z]^\top$, where u_x, u_y, and u_z denote the acceleration in the North-East-Down inertial frame. Each UAV is controlled by four control inputs $\mathbf{U} = [U_1, U_2, U_3, U_4]$ computed by the autopilot, for example, PX4 in our work, where U_1 is the thrust force along the vertical direction and U_2, U_3, and U_4 are rolling, pitching, and yawing moments, respectively. Aligned with MARL-related mainstream UAV studies for MC-PEGs [3], [32], we assume the UAV swarm flies at a fixed altitude, by constraining $u_z \equiv 0$. After obtaining the desired acceleration $\mathbf{u} = [u_x, u_y]$ in the horizontal direction by the proposed MARL method, it is sent to the autopilot, for example, PX4, to calculate the thrust force $\mathbf{U}$ based on the proportion integration differentiation (PID) algorithm [39], with which the physical simulator then iterates the UAV's pose based on the Newton-Euler formalism [40].

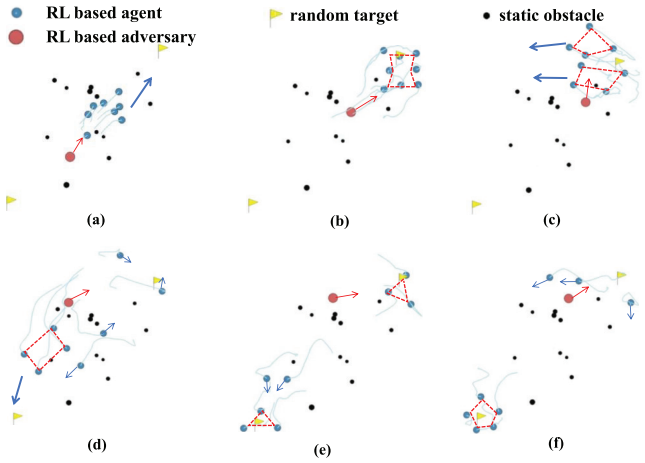


Fig. 1. Illustration of the CEFC task, wherein the UAV swarm (in blue dots) with some formation pattern is required to automatically respond to the adversary (e.g., wildfire, in the red circle) and obstacles (in the black circle) and move toward the multiple target areas (shown as the yellow flag) to carry out their missions (e.g., dropping supplies) in a communication-limited decentralized manner. (a) Randomly initialize and avoid obstacles. (b) Aggregate to the desired formation. (c) Split, change target, and transform into a new formation. (d) Escape and approach a new target. (e) Score on both targets. (f) New agents arrive and change formation adaptively.

2) CEFC Task Model: We primarily consider a CEFC task as illustrated in Fig. 1, wherein a set $\mathcal{N}$ of UAVs, with $|\mathcal{N}| = N$, are required to carry out their missions at target areas (i.e., formation coverage) in a communication-limited decentralized manner, as shown in Fig. 1(a) and (b). Notably, we denote $\mathcal{T}$ as the set of target areas, and each target area $\mathfrak{T} \in \mathcal{T}$ can be represented as $\mathfrak{T} = (\mathbf{p}_{\mathfrak{T}}, \kappa_{\mathfrak{T}}^{(t)})$, where $\mathbf{p}_{\mathfrak{T}}$ denotes the position and $\kappa_{\mathfrak{T}}^{(t)}$ denotes the urgency of the target area, which decreases as the agents arrive. Besides, the adversary $\mathfrak{A}$ should prevent the agents from reaching the target areas, and in this work, due to the algorithmic maturity, domain suitability for continuous control, and implementation practicality with reduced hyperparameter sensitivity [41], a default PPO policy is primarily used for the adversary to trace the nearest group.

In case of the possible adversary [see Fig. 1(c)] and obstacles [see Fig. 1(d) and (e)], the set of UAVs switch among a set of predefined formation patterns $\{\Delta_c | \forall c \in \mathcal{C}\}$, where c represents the agent quantity in formation Δ_c and $\mathcal{C}$ denotes the set of possible patterns. Hence, at each time step t, each agent i needs to recognize its $c-1$ teammates cooperatively and spontaneously determine one formation pattern c, resulting in $\chi^{(t)}$ groups with each group $\mathcal{N}_k, k \in \{1, \ldots, \chi^{(t)}\}$ of agents satisfying $|\mathcal{N}_k| = n_k$, n_0 isolated agents (i.e., no sufficient number of neighbors within the observation range to form any pattern in $\mathcal{C}$) and $n_0 + n_1 + \cdots + n_{\chi^{(t)}} = N$. In response to changes in the number of neighbor agents [see from 3 in Fig. 1(e) to 5 in Fig. 1(f)], the UAV shall switch the formation pattern automatically.

To accomplish the CEFC task, we formulate the problem as a decentralized partially observable Markov decision process (Dec-POMDP), which is defined as $\langle \mathcal{N}, \mathcal{S}, \mathcal{U}, P, \mathcal{Z}, \mathbf{R}, \Omega, \gamma \rangle$. In the CEFC task, $\mathcal{S}$ denotes the global state space while $\mathcal{U}$ is the homogeneous action space for a single agent. At each time step t, owing to the scant ability of perception against the colossal environment, each agent $i \in \mathcal{N}$ obtains a local state $\mathbf{z}_i^{(t)} \in \mathcal{Z}$ via the local state function $\Omega(\mathbf{z}_i^{(t)}|\mathbf{s}^{(t)}, i)$:$\mathcal{S} \times \mathcal{N} \times \mathcal{Z} \to [0, 1]$ instead of the state $\mathbf{s}^{(t)} \in \mathcal{S}$ at each time step and adopts an action $\mathbf{u}_i^{(t)} \in \mathcal{U}$ according to the individual policy $\boldsymbol{\pi}_i(\cdot|\mathbf{z}_i^{(t)})$:$\mathcal{Z} \times \mathcal{U} \to [0, 1]$. The joint action $\mathbf{u}^{(t)} = [\mathbf{u}_1^{(t)}, \mathbf{u}_2^{(t)}, \ldots, \mathbf{u}_N^{(t)}]$ taken at the current state $\mathbf{s}^{(t)}$ makes the environment transit into the next state $\mathbf{s}^{(t+1)}$ according to the function $P(\mathbf{s}^{(t+1)}|\mathbf{s}^{(t)}, \mathbf{u}^{(t)})$:$\mathcal{S} \times \mathcal{U} \times \mathcal{S} \to [0, 1]$. All agents share a global reward function $\mathbf{R}(\mathbf{s}^{(t)}, \mathbf{u}^{(t)})$:$\mathcal{S} \times \mathcal{U} \to \mathbb{R}$ with a discount factor γ and agents need to maximize the discounted accumulated reward $\mathbb{E}\left[\sum_t \gamma^t \mathbf{R}^{(t)}\right]$. Consistent with the Dec-POMDP framework, we specify the elements as follows.

1) *Local State:* The local state $\mathbf{z}_i^{(t)}$ should encompass the information of neighbors, target areas, and the adversary. For brevity, we denote the relative position and velocity of agent j with respect to i as $\mathbf{p}_{i\to j}$ and $\mathbf{v}_{i\to j}$. The observation of target areas and the adversary can be denoted as $\mathbf{o}_{\text{tar},i}^{(t)} = \{\mathbf{p}_{i\to\mathfrak{T}}^{(t)}, \kappa_{\mathfrak{T}}^{(t)} | \mathfrak{T} \in \mathcal{T}\}$ and $\mathbf{o}_{\text{adv},i}^{(t)} = [\mathbf{p}_{i\to\mathfrak{A}}^{(t)}, \mathbf{v}_{i\to\mathfrak{A}}^{(t)}]$, respectively. Meanwhile, agent i obtains some direct observation $\mathbf{o}_{\text{nei},i}^{(t)} = \{\mathbf{p}_{i\to j}^{(t)}, \mathbf{v}_{i\to j}^{(t)} | \forall j \in \boldsymbol{\xi}_i^{(t)}\}$ of their neighbors through communications and receives exchanged messages $\mathbf{M}_{\text{nei},i}^{(t)} = \{\mathbf{m}_j^{(t)} | \forall j \in \boldsymbol{\xi}_i^{(t)}\}$,[1] where $\mathbf{m}_j^{(t)}$ is a learnable vector to be communicated and $\boldsymbol{\xi}_i^{(t)}$ denotes the set of agents satisfying that the Euclidean distance $\|\mathbf{p}_{i\to j}^{(t)}\|$ is less than a maximum observation distance δ_{obs} (i.e., $\|\mathbf{p}_{i\to j}^{(t)}\| < \delta_{\text{obs}}$). For the sake of simplicity, the local observation of agent i is defined as $\mathbf{o}_i^{(t)} = [\mathbf{o}_{\text{tar},i}^{(t)}, \mathbf{o}_{\text{adv},i}^{(t)}, \mathbf{o}_{\text{nei},i}^{(t)}]$. Besides, the detection results $\mathbf{d}_i^{(t)} = [d_{i1}^{(t)}, \ldots, d_{iM}^{(t)}]$ of M light detection and ranging (LiDARs) [10] with angle resolution $2\pi/M$ are also used as part of the local state to help the agent avoid obstacles. Thus, the local state of agent i is summarized as $\mathbf{z}_i^{(t)} = [\mathbf{o}_i^{(t)}, \mathbf{M}_{\text{nei},i}^{(t)}, \mathbf{d}_i^{(t)}]$.
2) *Action:* As mentioned in Section III-A1, based on the local state $\mathbf{z}_i^{(t)}$, each agent sets its acceleration $\mathbf{u}_i^{(t)} = [u_{x_i}^{(t)}, u_{y_i}^{(t)}] \in \mathcal{U}$ following policy $\boldsymbol{\pi}_i(\cdot|\mathbf{z}_i^{(t)})$ individually to complete the CEFC task.
3) *Reward Function:* For the CEFC task, the reward function is designed to summarize multiple ingredients with weights ω_f, ω_n, ω_t, ω_e, and ω_c as

$$\mathbf{R}^{(t)} = \omega_f R_f^{(t)} + \omega_n R_n^{(t)} + \omega_t R_t^{(t)} + \omega_e R_e^{(t)} + \omega_c R_c^{(t)} \quad (1)$$

where the Hausdorff distance (HD)-based [42] formation reward $R_f^{(t)}$ measures the topological distance between the current and the expected formation for each group, while the navigation reward $R_n^{(t)}$, weighted by the urgency of target areas, incentivizes agents to efficiently reach target areas. The task accomplishment reward $R_t^{(t)}$ is used to quantify the progress of the formation coverage. The evasion reward $R_e^{(t)}$ and collision avoidance reward $R_c^{(t)}$, which serve as a critical safety mechanism, penalize agents in close proximity to the adversary and obstacles. We leave their specific expressions in Appendix A.

[1]The detailed procedure to acquire these messages shall be discussed in Section IV-C.

B. Multiagent Proximal Policy Optimization

To guide all agents to maximize the discounted accumulated reward $\mathbb{E}\left[\sum_t \gamma^t \mathbf{R}^{(t)}\right]$, MAPPO [16], which combines the single-agent PPO [15] and CTDE to learn the policy $\boldsymbol{\pi}_{\theta_i}(\cdot|\mathbf{z}_i^{(t)})$ $(\forall i)$ and value function $V_\phi(\mathbf{s}^{(t)})$:$\mathcal{S} \to \mathbb{R}$ parameterized by θ_i and ϕ, is used. Consistent with PPO, MAPPO maintains the old-version $\theta_{i,\text{old}}$ and ϕ_{old} and uses $\theta_{i,\text{old}}$ to interact with the environment and accumulate samples. Furthermore, θ_i and ϕ are periodically updated to maximize

$$\begin{aligned} J_{\pi_i}^{(t)}(\theta_i) &= \min\left(\beta_i^{(t)} \hat{A}^{(t)}, \text{clip}\left(\beta_i^{(t)}, 1-\varepsilon, 1+\varepsilon\right) \hat{A}^{(t)}\right) \\ J_V^{(t)}(\phi) &= -\left(V_\phi\left(\mathbf{s}^{(t)}\right) - \left(\hat{A}^{(t)} + V_{\phi_{\text{old}}}\left(\mathbf{s}^{(t)}\right)\right)\right)^2 \end{aligned} \quad (2)$$

where $\beta_i^{(t)} = (\boldsymbol{\pi}_{\theta_i}(\mathbf{u}_i^{(t)}|\mathbf{z}_i^{(t)}))/(\boldsymbol{\pi}_{\theta_{i,\text{old}}}(\mathbf{u}_i^{(t)}|\mathbf{z}_i^{(t)}))$, the clipping function $\text{clip}(\beta_i^{(t)}, 1-\varepsilon, 1+\varepsilon)$ constrains $\beta_i^{(t)} \in [1-\varepsilon, 1+\varepsilon]$, and $\hat{A}^{(t)} = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta^{(t+l)}$ is the generalized advantage estimation (GAE) [43] with $\delta^{(t)} = \mathbf{R}^{(t)} + \gamma V_{\phi_{\text{old}}}(\mathbf{s}^{(t+1)}) - V_{\phi_{\text{old}}}(\mathbf{s}^{(t)})$. Notably, ε and λ are the hyperparameters. Thus, the final optimization objective of MAPPO can be given by

$$J_{\text{MAPPO}} = \mathbb{E}_{i,t}\left[J_{\pi_i}^{(t)}(\theta_i) + J_V^{(t)}(\phi) + \alpha \mathcal{H}\left(\pi_{\theta_i}\left(\cdot|\mathbf{z}_i^{(t)}\right)\right)\right] \quad (3)$$

where α is coefficient and $\mathcal{H}$ is the entropy function.

C. Problem Formulation

We expect the MARL-driven agents to minimize the urgency of the target areas while avoiding collisions with obstacles and the PPO-driven adversary. Meanwhile, the parameters of each agent are shared during training to improve the learning efficiency, which implies $\boldsymbol{\pi}_{\theta_i} = \boldsymbol{\pi}_\theta$ $(\forall i \in \mathcal{N})$. Accordingly, we propose a system utility function $\mathfrak{J}$ as the objective of policy optimization, which can be expressed as

$$\max_{\boldsymbol{\pi}_\theta} \mathfrak{J} = \max_{\boldsymbol{\pi}_\theta} \mathbb{E}_t\left[\mathbf{R}^{(t)} \middle| \boldsymbol{\pi}_\theta\left(\cdot|\mathbf{z}^{(t)}\right)\right]. \quad (4)$$

Basically, we follow the CTDE architecture and leverage MAPPO for multiagent control. Recalling that $\mathbf{z}_i^{(t)} = [\mathbf{o}_i^{(t)}, \mathbf{M}_{\text{nei},i}^{(t)}, \mathbf{d}_i^{(t)}]$ $(\forall i)$, the communicated information (i.e., $\mathbf{m}_i^{(t)}$) shall impact the final performance. Meanwhile, classical solutions to CEFC rely on a centralized leader and cannot be directly applied to distributed agents. Therefore, to train a policy that can be executed in a completely distributed manner, we resort to a consensus inference module ConsMAC to compute $\mathbf{m}_i^{(t)}$ in Section IV-C and propose a distillation-based, agent number-adaptable distributed solution.

IV. Consensus Inference-Based HRL

A. Overview of the CI-HRL Framework

In this section, we present an overview of the proposed CI-HRL framework, which capably addresses the challenges of cooperative evasion from the adversary and the execution of formation coverage tasks under the constraints of limited communication. As shown in Fig. 2, CI-HRL implements a decentralized, real-time inference of the environment, and enables the UAV swarm to dynamically split and reorganize to accomplish the CEFC task. Specifically, for each agent i, CI-HRL automatically determines the anchor point $\mathbf{p}_{a,i} \in \mathcal{P}_a$

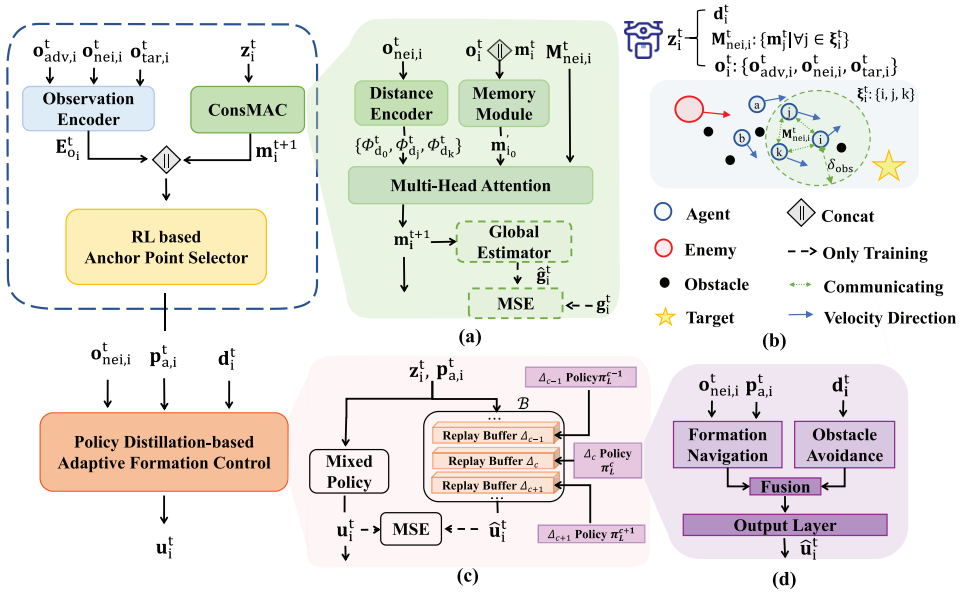


Fig. 2. Overview of CI-HRL. High-level policy (see Section IV-C) generates anchor point $\mathbf{p}_{a,i}^{(t)}$. (a) ConsMAC Module: Process local state $\mathbf{z}_i^{(t)}$ to infer global state $\mathbf{g}_i^{(t)}$ and generate the consensus message $\mathbf{m}_i^{(t+1)}$. (b) Communication process for $\mathbf{M}_{\text{nei},i}^{(t)}$ under a limited range. Low-level policy (see Section IV-B) outputs specific actions $\mathbf{u}_i^{(t)}$ for task execution. (c) Policy distillation process for adaptive formation control. (d) AT-M for single-formation policy.

as the temporary target location over a specified duration, where $\mathcal{P}_a$ denotes a candidate set of discrete or continuous coordinates. Hence, to strike a balance between evasion and mission execution, a subset of UAVs may select anchor points proximal to target areas for formation coverage, while others choose anchor points at a distance from predators to evade.

The whole training process consists of two stages, and the final policy can be represented as a combination of two-level policies, $\boldsymbol{\pi}_\Theta = [\boldsymbol{\pi}_L, \boldsymbol{\pi}_H]$, where Θ denotes the collective parameters of multiple DNNs. During the low-level stage, each agent i should learn a policy $\boldsymbol{\pi}_L(\mathbf{u}_i|\mathbf{p}_{a,i}, \mathbf{z}_i)$, which is a distillation-based adaptive formation solution AT-M, to output continuous acceleration $\mathbf{u}_i$ to control its journey to a specific anchor point $\mathbf{p}_{a,i}$ while maintaining formation with its neighbors. Then, at the high-level stage, a ConsMAC-based, decentralized high-level policy $\boldsymbol{\pi}_H(\mathbf{p}_{a,i}|\mathbf{z}_i)$ is trained to output the anchor point for guiding the low-level policy, thus jointly accomplishing the CEFC task. It should be noticed that the low-level policy solely accounts for formation with neighbors and navigation toward the anchor point, implying that when some UAVs select congruent anchor points that significantly differ from the rest of the swarm, these UAVs will automatically secede from the main group to form a new subgroup. Consequently, the formation pattern, to which each UAV belongs, is indirectly determined by the high-level decision-making. To simplify the following presentation, we denote a multilayer perceptron (MLP)-based DNN as $\mathcal{F}(\cdot)$.

B. Low-Level Policy

In this section, we design an effective low-level policy that allows agents to constitute some predefined formations and move toward the anchor point in a communication-limited, decentralized manner. Notably, we start with a low-level policy for a specific formation while ignoring the target areas as well as the adversary in Section IV-B2. During the training, all agents belong to the same group (i.e., $\chi^{(t)}$ is set to 1). Afterward, more general cases for flexible formation patterns will be investigated in Section IV-B3.

1) Low-Level Objective: As mentioned earlier, the low-level policy considers how to avoid obstacles and travel to the anchor point with formation, yielding the acceleration control of the agent. Therefore, the input only includes the observation of neighbors $\mathbf{o}_{\text{nei},i}^{(t)}$, LiDAR detection results $\mathbf{d}_i^{(t)}$, and the anchor point $\mathbf{p}_a$, which during the training are the same and randomly given by the environment in each episode. In Section IV-C, we will explain how to compute appropriate anchor points for practical execution. Meanwhile, since the low-level policy focuses more on the navigation and formation, the task and evasion reward (i.e., R_t and R_e) in (1) are ignored in this subsection. Therefore, the low-level optimization objective can be simplified as $\max_{\boldsymbol{\pi}_L} \mathfrak{I}_L = \max_{\boldsymbol{\pi}_L} \mathbb{E}_t[\mathbf{R}_L^{(t)}|\boldsymbol{\pi}_L]$, where $\mathbf{R}_L^{(t)} = \omega_f R_f^{(t)} + \omega_n R_n^{(t)} + \omega_c R_c^{(t)}$. Moreover, given that we focus on anchor points instead of target areas, the navigation reward given by (16) in Appendix A-2 can be regarded as $R_n^{(t)} = -\|\bar{\mathbf{p}}^{(t)} - \mathbf{p}_a\|$, where $\bar{\mathbf{p}}^{(t)}$ is the center of agents.

2) AT-M for Single-Formation Policy: Despite the remarkable performance in small-scale, fully connected multiagent formation tasks, MAPPO [16] in Section III-B faces significant difficulties in training formation and navigation policy directly in a partially observable environment with random obstacles. For example, our results show that by assigning a small weight ω_c to collision avoidance reward, MAPPO converges to a well-formed policy with frequent collisions. On the contrary, the policy ends up with poor formation competence for enlarging ω_c. Motivated by these facts, we adopt an AT-M to fuse multiple coupled constraints and better balance the tradeoff therein.

Beforehand, consistent with the conventional MAPPO framework, the corresponding value function V_{ϕ_L} is implemented by using an MLP [16]. As shown in Fig. 2(d), for a fixed formation $\{\Delta_c|\forall c \in \mathcal{C}\}$, AT-M aims to obtain a policy network $\boldsymbol{\pi}_L^c$ parameterized by $\Theta_L^c = [\theta_{\text{Lf}}^c, \theta_{\text{La}}^c, \theta_{\text{Lo}}^c]$, which consists of three MLP-based parts, namely, the formation and navigation module $\mathcal{F}_{\theta_{\text{Lf}}^c}$, the obstacle avoidance module $\mathcal{F}_{\theta_{\text{La}}^c}$ and the output layer $\mathcal{F}_{\theta_{\text{Lo}}^c}$. In particular, for each time step t,

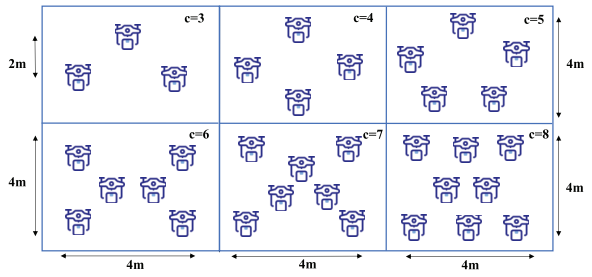


Fig. 3. Illustration of the predefined formations corresponding to different numbers of UAVs (i.e., $c \in \mathcal{C} = \{3, 4, 5, 6, 7, 8\}$).

$\mathcal{F}_{\theta^c_{\text{Lo}}}$ is used to sample an action $\mathbf{u}_i^{(t)} \sim \text{Normal}(\boldsymbol{\mu}_i^{(t)}, \boldsymbol{\sigma}_i^{(t)})$ as the final output by computing the mean and variance of the Gaussian distribution as

$$\boldsymbol{\mu}_i^{(t)}, \boldsymbol{\sigma}_i^{(t)} = \mathcal{F}_{\theta^c_{\text{Lo}}}\left(\mathcal{F}_{\theta^c_{\text{Lf}}}\left(\mathbf{p}_a, \mathbf{o}^{(t)}_{\text{nei},i}\right) + \mathcal{F}_{\theta^c_{\text{La}}}\left(\mathbf{d}_i^{(t)}\right)\right). \tag{5}$$

AT-M first independently trains DNNs parameterized by Θ_L^c in an environment without and with the involvement of obstacles (i.e., some $\omega_c \neq 0$) and obtains two sets of corresponding parameters $\Theta^c_{L,1}$ and $\Theta^c_{L,2}$, respectively. On this basis, AT-M fine-tunes a merged policy parameterized by $[\theta^c_{\text{Lf},1}, \theta^c_{\text{La},2}, \theta^c_{\text{Lo},1}]$ in the original environment to achieve the policy $\boldsymbol{\pi}_L^c$. Correspondingly, the swarm is endowed with the capacity to reach the anchor point in a fixed formation Δ_c while avoiding obstacles. Due to the independent pretraining of modules across disparate environments, we regard this approach as an alternative training.

3) Policy Distillation-Based Adaptive Formation Control: Following Section IV-B2, we can obtain a set of policies $\{\boldsymbol{\pi}_L^c | \forall c \in \mathcal{C}\}$, respectively, for formation Δ_c depicted in Fig. 3. In this part, we regard these policies as teacher models (i.e., a teacher model $\boldsymbol{\pi}_L^c$ instructs the formation Δ_c) and utilize policy distillation [44] to obtain a mixed-formation policy, to reduce local memory occupation. As shown in Fig. 2(c), we collect both inputs and outputs of policies $\{\boldsymbol{\pi}_L^c | \forall c \in \mathcal{C}\}$ by constituting a replay memory $\mathcal{B} = \{\langle \mathbf{p}_a^c, \mathbf{z}^c, \mathbf{u}^c \rangle_{\times \Lambda} | \forall c \in \mathcal{C}\}$, where $\mathbf{u}^c$ is generated by a learned teacher model $\boldsymbol{\pi}_L^c$ to form Δ_c and Λ is the capacity of replay buffer. Considering the dimension of observation $\mathbf{z}^c$ is determined by the agent number c, we align the vectors of observations in different formations by a zero-padding operation. In each training episode, memories (i.e., $\mathbf{z}$) from different teacher models are fed into the student model $\boldsymbol{\pi}_L$ simultaneously to calculate the corresponding $\hat{\mathbf{u}} = \boldsymbol{\pi}_L(\mathbf{p}_a, \mathbf{z})$. Afterward, we train $\boldsymbol{\pi}_L$ by minimizing the mean-squared-error (mse) loss

$$\mathcal{L}_{\text{PD}}(\Theta_L) = \sum_{\mathbf{u} \in \mathcal{B}} \|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 \tag{6}$$

where Θ_L is the parameter of $\boldsymbol{\pi}_L$. After updating Θ_L through (6), the mixed-formation policy set $\{\boldsymbol{\pi}_L^c | \forall c \in \mathcal{C}\}$ is merged to one student policy $\boldsymbol{\pi}_L$, which significantly saves the usage of agents' memory.

To sum up, we describe the training procedure of the low-level policy in Algorithm 1.

Algorithm 1 Training of Low-Level Policy

1: Initialize the set of quantities $\mathcal{C}$.
// AT-M for each formation pattern
2: **for** each $c \in \mathcal{C}$ **do**
3: **// Step 1**
4: Initialize the environment corresponding to c;
5: Train the policy and value function parameterized by $\Theta^c_{L,1} = [\theta^c_{\text{Lf},1}, \theta^c_{\text{La},1}, \theta^c_{\text{Lo},1}]$ and $\phi^c_{L,1}$ with random initialization by MAPPO;
6: **// Step 2**
7: Initialize the environment with random obstacles;
8: Train the policy and value function parameterized by $\Theta^c_{L,2} = [\theta^c_{\text{Lf},2}, \theta^c_{\text{La},2}, \theta^c_{\text{Lo},2}]$ and $\phi^c_{L,2}$ with random initialization by MAPPO;
9: **// Step 3**
10: Initialize the environment with random obstacles;
11: Fine-tune the policy and value function parameterized by $\Theta^c_L = [\theta^c_{\text{Lf},1}, \theta^c_{\text{La},2}, \theta^c_{\text{Lo},1}]$ and $\phi^c_{L,2}$ by MAPPO;
12: **end for**
// Policy Distillation
13: Initialize the replay memory $\mathcal{B} \leftarrow \varnothing$ and the student policy $\boldsymbol{\pi}_L$ with random parameters Θ_L, the training batch size N_b;
14: Collect tuples $\langle \mathbf{p}_a, \mathbf{z}, \mathbf{u} \rangle$ in each environment by $\{\boldsymbol{\pi}_L^c | \forall c \in \mathcal{C}\}$, respectively, and store them in $\mathcal{B}$;
15: **for** each policy distillation epoch **do**
16: Sample a batch of N_b tuples from $\mathcal{B}$;
17: Calculate $\hat{\mathbf{u}}$ based on $\mathbf{p}_a$ and $\mathbf{z}$;
18: Update Θ_L according to (6) via gradient descent and Adam optimizer;
19: **end for**
Output: The trained Θ_L;

C. High-Level Policy

Contingent on the low-level policy $\boldsymbol{\pi}_L$ for yielding the formation and obstacle avoidance action $\mathbf{u}$, the MAPPO-based high-level policy $\boldsymbol{\pi}_H$ can avoid the cumbersomeness of considering underlying tasks and put more emphasis on selecting an appropriate anchor point $\mathbf{p}_a$, which fully takes account of evading the adversary for accomplishing the CEFC task. Similar to the low-level policy, the corresponding value function V_{ϕ_H} is implemented by an MLP as well. However, different from MAPPO, as illustrated in Fig. 2(a), the high-level policy introduces a novel, supervised-learning-based consensus inference method ConsMAC, which can provide global consensus and complement the RL-based target selector to calculate certain decisions from a more global perspective.

1) High-Level Objective: Since the low-level policy saves the high-level policy from the formation and collision avoidance, the optimization objective of high-level policy can be represented as $\max_{\boldsymbol{\pi}_H} \Im_H = \max_{\boldsymbol{\pi}_H} \mathbb{E}_t[\mathbf{R}_H^{(t)} | \boldsymbol{\pi}_H, \boldsymbol{\pi}_L]$, where $\mathbf{R}_H^{(t)} = \omega_n R_n^{(t)} + \omega_t R_t^{(t)} + \omega_e R_e^{(t)}$. Recalling $\mathbf{z}_i^{(t)} = [\mathbf{o}_i^{(t)}, \mathbf{M}^{(t)}_{\text{nei},i}, \mathbf{d}_i^{(t)}]$, the maximization of this high-level objective lies in the effectiveness of exchange messages $\mathbf{M}^{(t)}_{\text{nei},i} = \{\mathbf{m}_j^{(t)} | \forall j \in \boldsymbol{\xi}_i^{(t)}\}$.

2) ConsMAC-Based Anchor Point Selection: Targeted at effectively aggregating observations $\mathbf{o}_i$ and messages $\mathbf{M}_{\text{nei},i}$ received from neighbors, ConsMAC is carefully calibrated and encompasses the following parts. First, to incorporate historical information, each agent i adopts a GRU-alike memory module $\mathcal{F}_{\psi_M}$ parameterized by ψ_M to process the concatenation of its own message $\mathbf{m}_{i_0}^{(t)}$ and the local observation $\mathbf{o}_i^{(t)}$, that is,

$$\mathbf{m}'_{i_0} = \mathcal{F}_{\psi_M}\left(\left[\mathbf{m}_{i_0}^{(t)} \middle|\middle| \mathbf{o}_i^{(t)}\right]\right) \tag{7}$$

where $\|$ represents the concatenation operation and the dimensions of both $\mathbf{m}_{i_0}^{(t)}$ and $\mathbf{m}'_{i_0}$ are the same. Meanwhile, for simplicity of representation, for each agent $i \in \mathcal{N}$, i_j denotes the jth nearest neighbor while i_0 indicates itself. Next, in resemblance to the positional encoding in Transformer [45], we use a learnable distance encoder to distinguish messages from agents with different distances and correspondingly assign different attention weights. Mathematically,

$$\mathbf{E}_{m_i}^{(t)} = \left[\mathbf{m}'_{i_0} \| \Phi_{d_0}^{(t)}\right], \quad \mathbf{E}_{m_{-i}}^{(t)} = \left[\mathbf{m}_{i_1}^{(t)} \middle\| \Phi_{d_1}^{(t)}, \ldots, \mathbf{m}_{i_k}^{(t)} \middle\| \Phi_{d_k}^{(t)}\right]^\top \tag{8}$$

where $\Phi_{d_j}^{(t)} = \sqrt{(1/D)}\ [\cos(w_1\|\mathbf{p}_{i\to i_j}^{(t)}\|), \ldots, \cos(w_D\|\mathbf{p}_{i\to i_j}^{(t)}\|)\ ]^\top$, $\psi_D = [w_1, \ldots, w_D]$ are the trainable parameters, D is the dimension of the latent space and $k = |\boldsymbol{\xi}_i^{(t)}|$ denoting the number of neighbors. Then, each agent can aggregate a latent vector $\mathbf{m}_i^{(t)}$ as the message for time step $t+1$ by

$$\mathbf{m}_i^{(t+1)} = \mathrm{MHA}_{\psi_A}\left(\mathbf{E}_{m_i}^{(t)}, \mathbf{E}_{m_{-i}}^{(t)}, \mathbf{E}_{m_{-i}}^{(t)}\right) \tag{9}$$

where MHA is a multihead attention layer [45] parameterized by ψ_A. Furthermore, we infer the global information by a global estimator $\mathcal{F}_{\psi_E}$ parameterized by ψ_E, and the estimated state embedding can be written as

$$\hat{\mathbf{g}}_i^{(t)} = \mathcal{F}_{\psi_E}\left(\mathbf{m}_i^{(t+1)}\right). \tag{10}$$

We leverage global state $\mathbf{g}_i^{(t)} \in \mathbf{s}^{(t)}$ as the label for supervised learning and adopt mse as the loss function. Therefore, the loss function of ConsMAC can be formulated as

$$\mathcal{L}_{\mathrm{ConsMAC}}(\Psi) = \mathbb{E}_{i,t}\left[\left\|\hat{\mathbf{g}}_i^{(t)} - \mathbf{g}_i^{(t)}\right\|^2\right] \tag{11}$$

where $\Psi = [\psi_D, \psi_M, \psi_A, \psi_E]$. Notably, the specific choice of global state $\mathbf{g}_i^{(t)}$ could be rather flexible such as the anchor points of all agents $\mathbf{g}_i^{(t)} = \{\mathbf{p}_{a,j}^{(t)} | \forall j \in \mathcal{N}\}$ or the observations of all agents $\mathbf{g}_i^{(t)} = \{\mathbf{p}_{i\to j}^{(t)}, \mathbf{v}_{i\to j}^{(t)} | \forall j \in \mathcal{N}\}$, denoted as ConsMAC-A and ConsMAC-O, respectively. We evaluate both methods in Section V and validate the scalability of ConsMAC. In this way, minimize (11) ensures $\mathcal{F}_{\psi_E}(\mathbf{m}_i^{(t+1)}) \to \mathbf{g}_i^{(t)}$, and the intermediate output of ConsMAC in the local perspective can implicitly embed the global information of all agents (i.e., $\mathbf{m} = \mathcal{F}^{-1}(\mathbf{g}, \mathbf{o})$), rather than merely aligning local observations in methods like NVIF [11] (i.e., $\mathbf{m} \approx \mathcal{F}^{-1}(\mathbf{o})$). Due to the uniqueness of the global state, such a procedure can be interpreted as the establishment of consensus among neighbors.

Accordingly, the RL-based policy, which encompasses an observation encoder and an RL-based selector, yields a suitable anchor point corresponding to the observations and established consensus (i.e., $\mathbf{o}_i^{(t)}$ and $\mathbf{m}_i^{(t+1)}$). In particular, the embedding vector $\mathbf{E}_{o_i}^{(t)}$ can be obtained by the MLP-based observation encoder $\mathcal{F}_{\theta_{\mathrm{HO}}}$ and the RL-based selector $\mathcal{F}_{\theta_{\mathrm{HS}}}$ is used to calculate and sample the anchor point as the final output

$$\mathbf{p}_{a,i}^{(t)} \sim \mathcal{F}_{\theta_{\mathrm{HS}}}\left(\mathbf{E}_{o_i}^{(t)}, \mathbf{m}_i^{(t+1)}\right). \tag{12}$$

As mentioned in Section IV-A, we can treat $[\Theta_H, \Psi]$ as the parameters of the high-level policy $\boldsymbol{\pi}_H$, where $\Theta_H = [\theta_{\mathrm{HO}}, \theta_{\mathrm{HS}}]$.

3) Training Techniques: In a nutshell, the loss function of the high-level policy can be summarized as

$$\mathcal{L}_{\mathrm{high}}(\Theta_H, \phi_H, \Psi) = -J_{\mathrm{high}}(\Theta_H) - J_V(\phi_H) + \mathcal{L}_{\mathrm{ConsMAC}}(\Psi). \tag{13}$$

The optimization of Ψ for ConsMAC is a standard supervised learning problem that can be solved by a gradient descent algorithm. Then, the RL-based module Θ_H utilizes the message $\mathbf{m}^{(t+1)}$, yielded by ConsMAC, as part of the local information for decision-making as in (12) and employs the gradient obtained by RL to optimize the policy. It should be noticed that even though the message $\mathbf{m}^{(t+1)}$ is used as the input of the policy module, the RL loss does not backpropagate to ConsMAC. In other words, ConsMAC is designed as an independent information processing module to provide the global consensus, and the anchor point selector needs to learn how to use it through the RL method. During the training process, both ConsMAC and RL-based modules are alternately updated. On the other hand, in our hierarchical architecture, the training of the high-level policy is based on the lower-level network, which means that specific decisions of the agents in the environment are given by the low-level policy. Therefore, it is necessary to derive how to compute the specific RL gradient of $\boldsymbol{\pi}_{\Theta_H}$ in this nested scenario. In this regard, the gradient of J_{high} will be given by the following theorem. For convenience, we omit the superscript (t) in this part and simplify $\mathbf{m}$ and $\mathbf{z}$ as $\mathbf{s}$. Therefore, the high- and low-level policies are denoted as $\boldsymbol{\pi}_H(\mathbf{p}_a|\mathbf{s})$ and $\boldsymbol{\pi}_L(\mathbf{u}|\mathbf{s}, \mathbf{p}_a)$, respectively. To align with the PPO-based CI-HRL framework, we assume the joint policy for the agent to interact with the environment (i.e., $\boldsymbol{\pi}_\Theta = [\boldsymbol{\pi}_L, \boldsymbol{\pi}_H]$) is a stochastic policy, which implies that the two policies are not deterministic simultaneously.

Theorem 1: Given the high-level RL-based policy $\boldsymbol{\pi}_H$ and the low-level policy $\boldsymbol{\pi}_L$, the gradient of the objective function $J_{\mathrm{high}}(\Theta_H)$ with respect to the variable Θ_H is

$$\begin{aligned}&\nabla_{\Theta_H} J_{\mathrm{high}}(\Theta_H)\\ &= \mathbb{E}_{\mathbf{s},\mathbf{u}\sim\boldsymbol{\pi}_L,\ \mathbf{p}_a\sim\boldsymbol{\pi}_H}\Big[\big(\alpha_H \nabla_{\Theta_H} \ln \boldsymbol{\pi}_H(\mathbf{p}_a|\mathbf{s}) + \alpha_L \nabla_{\mathbf{p}_a} \ln \boldsymbol{\pi}_L(\mathbf{u}|\mathbf{s}, \mathbf{p}_a)\\ &\qquad \nabla_{\Theta_H} \boldsymbol{\pi}_H(\mathbf{p}_a|\mathbf{s})\big)\, Q(\mathbf{s}, \mathbf{u})\Big]\end{aligned} \tag{14}$$

where Q is the state-action value function of $\boldsymbol{\pi}_L$, $\mathbf{u}$ is the joint action of all agents. α_H is set to 1 if the high-level policy is stochastic and 0 otherwise. α_L is associated with the low-level policy following the same binary convention.

Proof: See Appendix B in the Supplementary File. ∎

To boost the performance of the trained high-level policy, we treat the low-level policy as a deterministic policy and use a navigation-only low-level policy, which neglects the constraint of formation, to pretrain the stochastic high-level policy. Afterward, we fine-tune the high-level policy in the complete environment with the formation constraint. To sum up, the training procedure is summarized in Algorithm 2.

V. Experimental Results and Discussions

In this section, we evaluate our method in both the MPE [9] and an SITL simulation environment based on the robot operating system (ROS) system, Gazebo-Classic physics simulator [46], and PX4 autopilot [39] for quadrotor UAVs. Notably, different formation patterns correspond to different formation numbers, as shown in Fig. 3.

A. Experimental Settings

During the experiments, we utilize some common training techniques in MAPPO such as orthogonal initialization, gradient clipping, and value normalization, while the value functions of both the high- and low-level policies are implemented by a three-layer MLP with a hidden size of 128. The training data is collected in 20 threads simultaneously, and the update epoch of PPO is 15. The Adam optimizer is used with

Algorithm 2 Training of CI-HRL

Initialize ConsMAC, the RL-based module and value function with random parameters Ψ, Θ_H, ϕ_H, respectively;
Initialize the training batch size of ConsMAC N_b, and the replay memory $\mathcal{B}_{\mathrm{C}} \leftarrow \varnothing$;
Initialize the adversary $\mathfrak{A}$ and the set of target areas $\mathcal{T}$;
// Pretraining of high-level policy
for each train episode **do**
  Randomly initialize the environment state $\mathbf{s}^{(0)}$;
  // Training of RL-based module
  for $t = \{1, \ldots, T\}$ **do**
    Each agent i obtains $\mathbf{o}_i^{(t)}$ from $\mathbf{s}^{(t)}$ and receives $\mathbf{M}_{\mathrm{nei,i}}^{(t)}$ from neighbors;
    Calculate $\mathbf{m}_i^{(t+1)}$ by (9) and $\mathbf{p}_{a,i}^{(t)}$ by (12);
    Store $\langle \mathbf{o}_i^{(t)}, \mathbf{M}_{\mathrm{nei,i}}^{(t)}, \mathbf{p}_{a,i}^{(t)}, \mathbf{s}_i^{(t)} \rangle$ in $\mathcal{B}_{\mathrm{C}}$.
    Calculate the state value $V_{\phi_H}(\mathbf{s}^{(t)})$;
    Calculate actions $\mathbf{u}_i^{(t)}$ by a navigation-only policy;
    Execute actions, obtain the reward $\mathbf{R}_H^{(t)}$ and update state $\mathbf{s}^{(t)} \rightarrow \mathbf{s}^{(t+1)}$;
  end for
  Update Θ_H, ϕ_H according to (13) while incorporating the training techniques of MAPPO;
  // Training of ConsMAC
  for each update epoch of ConsMAC **do**
    Sample a batch of N_b tuples from $\mathcal{B}_{\mathrm{C}}$;
    Calculates $\hat{\mathbf{g}}_i^{(t)}$ by (7)–(10);
    Update Ψ according to (11) via Adam optimizer;
  end for
end for
// Training of low-level policy
Train the low-level policy $\boldsymbol{\pi}_L$ by Algorithm 1;
// Fine-tuning
Fine tuning the high-level modules Ψ, Θ_H, ϕ_H in the environment by step 4-20, but the actions $\mathbf{u}_i^{(t)}$ is calculated by the low-level policy $\boldsymbol{\pi}_L$.

Output: The trained Ψ, Θ_H, Θ_L;

a learning rate of 1×10^{-4}. Meanwhile, aligned with MARL-related mainstream UAV studies for MC-PEGs [3], [32], we focus on the 2-D experiments with a fixed altitude, where the coordinates of UAVs are randomly initialized in the range $[-2, 2]$ m along the x- and y-axes. Moreover, the key parameter settings of the environment are summarized in Nomenclature, and the specific settings of each level are as follows.

1) Low-Level Settings: In our experiments, the low-level policy is updated for 500 episodes, each of which has 100 time steps, and the anchor point is randomly initialized in the range $[-5, 5]$ m for both the x- and y-axes. The formation and navigation module, the obstacle avoidance module, and the output layer in (5) are all composed of a three-layer MLP with a hidden size of 128 for all $c \in \mathcal{C}$. In each step of AT-M, we set up different environments as mentioned in Section IV-B2 with obstacle densities of 0 and $3 \times 10^{-2}/\mathrm{m}^2$, respectively, for training. However, when the number of agents is small, direct training in an environment with obstacles already leads to satisfactory results, and recombining modules may reduce performance. Therefore, in subsequent performance comparison, we only focus on the pattern greater than 5.

TABLE I
KEY PARAMETER SETTINGS OF THE ENVIRONMENT

Parameters	Settings
Number of UAVs	$N = 8$
The set of possible quantities	$\mathcal{C} = \{3, 4, 5, 6, 7, 8\}$
Range of velocity per UAV (m/s)	$[-1, 1]$
Range of adversary velocity (m/s)	$[-0.75, 0.75]$
Observation distance (m)	$\delta_{\mathrm{obs}} = 3$
Reward weights of $\mathbf{R}_{\mathrm{L}}$	$(\omega_{\mathrm{f}}, \omega_{\mathrm{n}}, \omega_{\mathrm{c}}) = (15, 4, 100)$
Reward weights of $\mathbf{R}_{\mathrm{H}}$	$(\omega_{\mathrm{t}}, \omega_{\mathrm{n}}, \omega_{\mathrm{e}}) = (10, 0.1, 100)$
PPO Hyperparameter	$(\gamma, \varepsilon, \lambda) = (0.8, 0.2, 0.95)$

2) High-Level Settings: The high-level policy is updated for 1000 episodes during pretraining and 500 episodes during fine-tuning, each of which has 400 time steps, but anchor points are generated by the high-level policy every ten steps. The predator primarily uses a single-agent PPO policy, with the same network structure as the low-level policy of UAVs, and is set to chase the nearest group with at least three members and avoid obstacles. For convenience, the discrete coordinate is used for the high-level policy, and the set of possible anchor points in the x- and y-axes is $\{-8, 0, 8\}$ m, implying a 9-D high-level action space. The distance encoder consists of a linear layer, while the global estimator is implemented by a three-layer MLP, and the rest of the MLP-based modules have two layers. Besides, we implement a four-head attention layer, and the dimension of the message $\mathbf{m}$ and attention layer is 64, while the hidden size of other MLPs is 128. Moreover, after every 20 episodes of interaction, the collected data are utilized to train ConsMAC 5000 times with a batch size of 2048. We assume the number of target areas is set to 2 (i.e. $|\mathcal{T}| = 2$), with each area randomly selected from the set of coordinates $\{(-8, -8), (-8, 8), (8, -8), (8, 8)\}$ m.

B. Performance of Low-Level Policy

1) Evaluation Metrics: We evaluate the formation performance in terms of the average *reward* per time step $\mathbf{R}_L$, the *formation stability* $\mathbf{F}$, the *navigation efficiency* $\mathbf{N}$, and average *collision probability* $\mathbf{C}$. Specifically, the formation stability $\mathbf{F}$ counts the average time of formation maintenance per episode (i.e., the time when the HD-based formation error defined in Appendix A-1 is less than 1 m), while the formation accuracy complies with the required <1.5 m positioning accuracy in D2D communication [47]. For navigation tasks, the navigation efficiency $\mathbf{N}$ quantifies the average time staying in proximity to the destination (i.e., $\|\bar{\mathbf{p}}^{(t)} - \mathbf{p}_a\| < 1$ m), such a positional accuracy in swarms can significantly enhance payload deployment success rates [48]. Additionally, the average collision probability $\mathbf{C}$ evaluates obstacle avoidance and the average reward per time step $\mathbf{R}_L$ provides a comprehensive performance evaluation aligned with operational objectives.

2) Performance Comparison: To test the performance of the AT-M under more severe conditions with denser obstacles, we have increased the density of obstacles to $5 \times 10^{-2}/\mathrm{m}^2$. We evaluate the curriculum learning-based formation [18], denoted as CL-M, classical multi-agent deep deterministic policy gradient (MADDPG) [9], traditional optimal reciprocal collision avoidance (ORCA)-based formation ORCA-F [49], and our AT-M of each step in MPE. Table II shows the average results of 50 episodes of tests. To address safety-critical scenarios, we further propose AT-M-Safe, a conservative variant of AT-M. By slightly adjusting LiDAR parameters (instead of $d_{im}^{(t)}$,

TABLE II
PERFORMANCE COMPARISON OF AT-M WITH BASELINES

c	Method	$\mathbf{R}_{\mathrm{L}}\uparrow$	**F** (s) ↑	**N** (s) ↑	**C** (%) ↓
6	MADDPG [9]	-	58.12	4.26	-
	ORCA-F [49]	−92.55	24.52	45.06	0.40
	CL-M [18]	−90.79	48.80	38.94	0.78
	AT-M-Step 1	−424.45	**59.64**	**46.20**	11.40
	AT-M-Step 2	−96.56	17.7	35.50	0.86
	AT-M-Step 3	−96.00	49.72	36.66	0.78
	AT-M-Safe	**−46.28**	29.54	24.24	**0.36**
7	MADDPG [9]	-	0.84	**46.94**	-
	ORCA-F [49]	−118.10	15.74	35.50	0.48
	CL-M [18]	−113.69	29.72	15.28	0.74
	AT-M-Step 1	−660.84	**55.78**	40.26	19.46
	AT-M-Step 2	−125.84	8.14	28.40	1.08
	AT-M-Step 3	−94.32	40.24	24.26	0.74
	AT-M-Safe	**−55.61**	23.92	16.88	**0.46**
8	MADDPG [9]	-	0.02	49.30	-
	ORCA-F [49]	−195.23	7.74	28.52	0.66
	CL-M [18]	−121.90	29.30	34.08	1.30
	AT-M-Step 1	−506.48	**45.84**	**53.92**	13.20
	AT-M-Step 2	−266.94	0.10	1.68	2.56
	AT-M-Step 3	−106.94	41.52	36.90	1.14
	AT-M-Safe	**−72.57**	20.26	26.60	**0.62**

$d_{im}^{(t)}-0.2$ is used in (5) when the mth LiDAR of the UAV detects an obstacle), UAVs initiate precautionary avoidance measures earlier. The results of AT-M-Step 2 indicate that initializing a model in the case of dense obstacles leads to excessive collision penalties, hindering the learning of effective formation strategy, but it still maintains commendable obstacle avoidance capabilities. Furthermore, it can be observed that our proposed AT-M (i.e., AT-M-Step 3) outperforms other baselines in terms of overall performance balance when the number of agents increases, which demonstrates that introducing a well-trained obstacle avoidance model into a formation-capable model and fine-tuning the integrated model can significantly produce appealing performance. In addition, MADDPG [9] is also trained in an environment without obstacles, but even so, when pattern c is larger than 6, MADDPG can no longer learn effective strategies, and its training overhead is very large compared to other methods. Meanwhile, ORCA-F [49], which is based on ORCA and a fully connected leader-follower framework, can greatly reduce the probability of collisions. However, ORCA-F performs poorly when integrated with downstream formation and navigation tasks, and becomes overly conservative as the number of agents increases, leading to further performance degradation, which highlights its limitations. Meanwhile, since the collision avoidance reward in (21) penalizes proximity to obstacles for proactive avoidance, AT-M-Safe achieves superior collision avoidance than ORCA-F while maintaining formation and navigation capabilities, further validating the universality and flexibility of the AT-M methodology. Nevertheless, AT-M-Safe is conservative as well, with significantly degraded formation and navigation performance compared to AT-M-Step 3. Thus, it serves as a backup for dense obstacle environments, while AT-M-Step 3 suffices for subsequent experiments. Different from AT-M, the CL-M [18] involves gradually increasing obstacle density during training. Notably, the performance gap between AT-M and CL-M gradually increases as the number of agents increases, since the curriculum learning method gradually enhances obstacle perception and inadvertently leads to a partial and irreversible loss of formation ability. In contrast, our AT-M can achieve obstacle avoidance while maintaining the original formation ability by fusing models from different stages and fine-tuning.

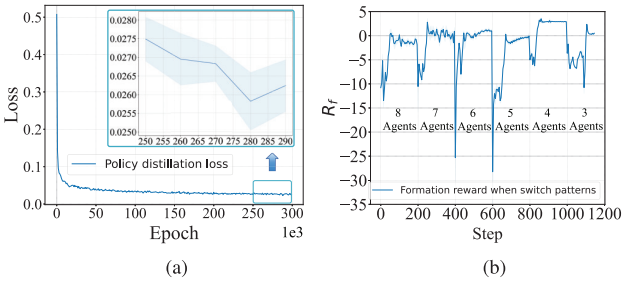


Fig. 4. Simulation results of adaptive formation. (a) Curve of loss during policy distillation. (b) Formation reward when a random agent drops out every 200 time steps.

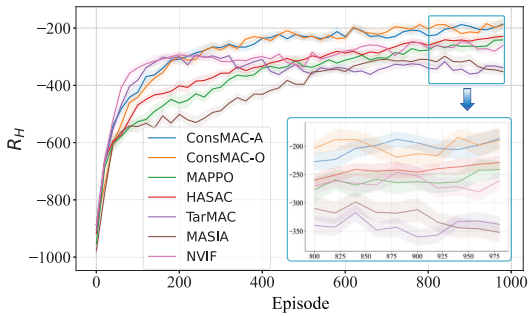


Fig. 5. Comparison to other communication methods.

3) Adaptive Formation Results: For the adaptive formation mentioned in Section IV-B3, we pretrain six formation models with the pattern c ranging from 3 to 8 in Fig. 3 and store the corresponding tuples $\langle \mathbf{p}_a, \mathbf{z}, \mathbf{u} \rangle$ for 60 000 time steps as the replay memory $\mathcal{B}$. Besides, the hidden size of the distilled model is 256 and the Adam optimizer is used with a learning rate of 1×10^{-4}. Meanwhile, the distilled student model is trained for 300 000 episodes, and the batch size of each episode of sampling is 600. The curve of loss during policy distillation is shown in Fig. 4(a), while Fig. 4(b) illustrates the curve of formation reward over time. In particular, given the initial existence of eight agents, some randomly selected UAVs are assumed to be no longer observed. In the adaptive formation task, agents are supposed to perceive the change in the fleet number itself and adapt the formation policy swiftly.

C. Performance of ConsMAC

1) Effectiveness and Superiority of ConsMAC: We compare our ConsMAC module with three representative MARL communication methods (i.e., the attention-based message aggregation method TarMAC [25], the supervised learning-based information extraction method MASIA [13], and the SOTA communication method NVIF [11]), and the communication content and overhead of each method are shown in Table III. In addition, we add the original MAPPO and the latest MARL SOTA algorithm HASAC [14] without communication to reflect the effect of the communication module. Fig. 5 presents the corresponding performance comparison while Table III lists the communication costs. As shown in Fig. 5, ConsMAC significantly outperforms other baselines, and it is worth noting that both ConsMAC-A and ConsMAC-O, which refer to training ConsMAC by using global outputs of anchor points or global observations of all agents as labels, respectively, significantly improve the overall performance,

TABLE III
COMMUNICATION OVERHEAD OF EACH METHOD

Method	Communication Content	Dimension
MAPPO [16]	-	0
HASAC [14]	-	0
TarMAC [25]	Latent Vector	64
MASIA [13]	Local Observation	38
NVIF [11]	Latent Vector + Local Observation	102
ConsMAC (Ours)	Latent Vector	64

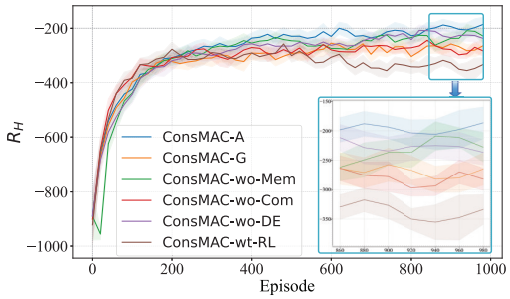


Fig. 6. Ablation results of ConsMAC-A.

demonstrating the robustness and generality of our method. On the other hand, TarMAC accelerates the convergence rate by introducing the implicit communication vector, but its final performance is not as good as the original MAPPO, possibly due to that it does not guide the communication content and produces invalid information. Besides, both the convergence rate and the final effect of MASIA are also reduced in the local communication environment, which indicates that only transmitting the original observation information cannot provide enough guidance for the agent to make decisions. Notably, NVIF converges as fast as TarMAC and ultimately outperforms both TarMAC and MASIA, demonstrating its improvements over existing communication methods. However, it still lags behind ConsMAC and MAPPO, indicating insufficient guidance in communication content. Meanwhile, when the number of training iterations aligns with other baselines, HASAC merely leads to significantly inferior results.

2) Ablation Study of ConsMAC-A: We perform the ablation study of ConsMAC-A to further demonstrate the effectiveness of each part in ConsMAC. Fig. 6 illustrates the results between ConsMAC-A and some variants. ConsMAC-wo-Com denotes the ConsMAC without communicated messages, while we remove the memory module and distance encoder in ConsMAC-wo-Mem and ConsMAC-wo-DE, respectively. Moreover, ConsMAC-wt-RL uses both supervised learning loss and RL loss to update ConsMAC. The results in Fig. 6 verify the individual contribution of each module. Meanwhile, ConsMAC-G uses the estimated state $\hat{\mathbf{g}}$ in (10) as the communicated message instead of $\mathbf{m}$, and the results show that the effect of communicating the hidden layer vector is significantly better than that of communicating $\hat{\mathbf{g}}$, and greatly contributes to the overall performance improvement. Besides, it is worth noting that adding the RL loss to the ConsMAC reduces the performance, which demonstrates the importance of independent learning in ConsMAC.

D. Performance of CI-HRL

1) MPE Simulation:

a) Effectiveness and superiority of CI-HRL: On the basis of ConsMAC-A, we follow Algorithm 2 and fine-tune the overall CI-HRL with the trained low-level policy. Besides, we directly train the high-level policy with the low-level policy for comparison, denoted as CI-HRL-DT, while we suffix CI-HRL methods without the last fine-tuning by "wo-FT." Fig. 7 compares the learning curves of both fine-tuning and direct training. Furthermore, in Table IV, we evaluate each method for 50 episodes and record the rewards for all parts, where **E** denotes the average time of dangerous situations per round (i.e., the distance between agents and the adversary is less than 2 m). The related results indicate that the high-level decision-making in CI-HRL significantly outperforms other MARL algorithms, achieving larger task rewards in target areas. Meanwhile, due to variations in the low-level strategies, a direct concatenation of these components cannot score more effectively. Nevertheless, the pretrained CI-HRL still performs

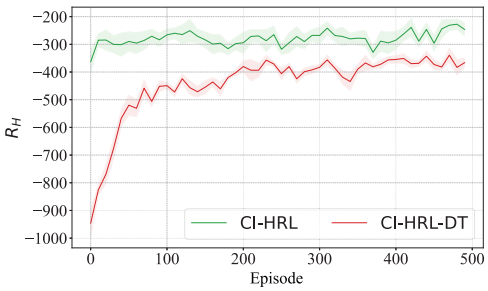


Fig. 7. Effectiveness of fine-tuning hierarchical policies in CI-HRL.

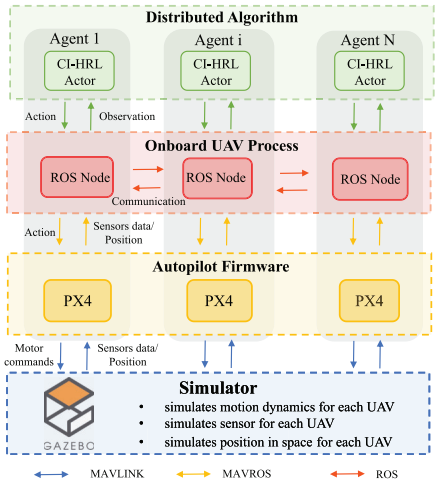


Fig. 8. Framework of SITL.

TABLE IV
PERFORMANCE COMPARISON OF CI-HRL WITH BASELINES

Method	$\mathbf{R}_{\mathrm{H}}\uparrow$	$R_{\mathrm{t}}\uparrow$	$R_{\mathrm{n}}\uparrow$	$R_{\mathrm{e}}\uparrow$	$\mathbf{E}\downarrow$
MAPPO [16]+AT-M	−397.29	54.33	−334.95	−116.68	8.78
HASAC [14]+AT-M	−381.07	105.70	−347.82	−138.95	11.42
TarMAC [25]+AT-M	−432.81	29.81	−367.06	−95.56	7.16
MASIA [13]+AT-M	−404.45	31.89	−336.17	−100.19	7.72
NVIF [11] +AT-M	−403.79	16.12	−340.58	−79.33	5.94
CI-HRL-w-CL-M-wo-FT	−408.55	38.81	−326.08	−121.29	9.48
CI-HRL-w-CL-M	−320.69	74.13	**−288.11**	−106.70	8.40
CI-HRL-DT	−366.95	79.72	−381.52	−65.16	5.16
CI-HRL-wo-FT	−351.27	58.31	−301.65	−107.93	7.66
CI-HRL (Ours)	**−281.56**	**107.37**	−327.27	**−61.66**	**4.46**

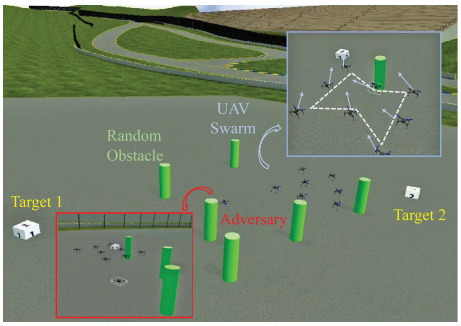


Fig. 9. Task overview in Gazebo simulator for SITL.

TABLE V
EVALUATION RESULTS OF DIFFERENT ADVERSARY STRATEGIES

	PPO	DDPG	R-Largest	R-Nearest
R_t	107.37	161.97	151.74	187.87
R_e	−61.66	−82.43	−55.48	−79.54
E	4.46	6.42	4.44	5.80

better on average reward and navigation efficiency than direct joint training. Moreover, the joint fine-tuning discussed in Section IV-C3 contributes to improving the performance. In addition, CL-M in Section V-B2 is used as the low-level policy of CI-HRL, and the resulting model CI-HRL-w-CL-M severely underperforms its AT-M counterpart, but the joint fine-tuning improves the overall performance significantly, demonstrating that an optimized high-level policy can compensate for the shortcomings of a less-performing low-level one.

b) Generalization results: To further validate the robustness and generalization of CI-HRL, we conduct extensive experiments under diverse adversary strategies and larger-scale scenarios, respectively. In addition to the original PPO-based adversary, we also evaluate the performance of the CI-HRL under DDPG-driven and two different rule-based adversary policies (i.e., tracing the largest group, denoted as R-Largest, and targeting the nearest agent, denoted as R-Nearest). Table V reveals the robust performance of CI-HRL against various adversary strategies. Notably, DDPG and R-Nearest strategies focus more on chasing the nearest agent rather than impeding the swarm's task completion, leading to lower evasion rewards. In other words, such an overemphasis on a single agent enhances the safety and task efficiency of the remaining agents, thereby improving the overall performance. Furthermore, we expand the number of agents to 15, and the agent-averaging results are shown in Table VI. Consistent with the trend in Table II, along with the increase in the number of agents, the formation performance of low-level policy decreases, which subsequently impacts the overall task completion. However, the average navigation and evasion rewards remain relatively stable, indicating that the high-level policy can make superior decisions to compensate for the low-level performance degradation. These results highlight the scalability and adaptability of our method to diverse adversarial dynamics and validate its potential for deployment in complex environments, motivating us to investigate the practical performance of CI-HRL.

2) SITL Simulation:

a) Framework of SITL: To verify the performance of the CI-HRL algorithm deployed on a quadrotor UAV in the real world, as shown in Fig. 8, we develop an ROS-based SITL simulation environment with Gazebo-Classic physics simulator and PX4 autopilot [39] for quadrotor UAVs. Different from existing open-source SITL work such as XT-Drone [50], in our simulation the UAV node makes fully distributed and asynchronous decisions based on partial observations within a limited communication range, and the CEFC task is simulated in high fidelity with multiple obstacles, target areas, and game scenario, as illustrated in Fig. 9. Specifically, each UAV in offboard control mode corresponds to an independent Python flight control process (CI-HRL algorithm with ConsMAC-A). Moreover, based on the MAVROS protocol, each UAV subscribes to/publishes ROS topics via user datagram protocol (UDP) connections to broadcast its states and observations regarding the adversary, obstacles, target areas, and neighboring agents. Then, each UAV calculates the expected acceleration according to the CI-HRL output and publishes the result to the topic subscribed by PX4. Meanwhile, PX4 gets the UAV's real-time pose and sensor messages via a transmission control protocol (TCP) connection with Gazebo. Based on the received acceleration requirement, PX4 calculates motor and actuator values from the PID controller and sends them to the Gazebo. Afterward, Gazebo determines the next frame's pose and sensor data according to the UAV's dynamic model and sends it back to PX4.

TABLE VI
GENERALIZATION RESULTS OF CI-HRL IN LARGE-SCALE GROUPS

N	$\mathbf{R}_H/N$	R_t/N	R_n/N	R_e/N	$\mathbf{E}/N$
8	−35.20	13.42	−40.91	−7.71	0.56
9	−36.80	15.67	−43.13	−9.34	0.68
10	−41.16	10.11	−43.61	−7.66	0.60
12	−42.26	8.91	−44.97	−6.20	0.49
15	−45.31	4.09	−41.46	−7.95	0.63

b) Strategy analysis: We depict four typical cooperation steps in one episode of SITL and the corresponding partial motion path based on the pose history from the ROS topic in Fig. 10. Besides, to verify how CI-HRL works, we save the 64-dimensional message $\mathbf{m}_i^{(t)}$ that is transmitted to neighboring agents. Furthermore, we compute the cosine similarity of the messages from UAV $\{1, 2, 3, 4\}$ at each high-level decision step to analyze the target intention of agents, as an illustration in Fig. 11(a). Notably, higher cosine similarity of messages implies more similar intention and consistent anchor points, since the communicated messages $\mathbf{m}_i^{(t)}$ imply the prediction of the global state as (10) and the selection of anchor points largely depends on the aggregated message $\mathbf{m}_i^{(t+1)}$ followed by the CI-HRL as (12). It can be observed from Fig. 10(a) that at the beginning of the simulation (i.e., step 1), given the existence of the adversary, all agents take unanimous decisions toward the top right corner target rather than the target $\mathbf{p}_1 = (-8, -8)$. The similarity matrix shown in Fig. 11(b) suggests all the message similarities among four agents are higher than 0.9, standing for agents' consensus on the target decision at the moment. At step 19, as shown in Fig. 10(b), due to the adversary's approach, the corresponding agents, which originally formed in Δ_8, undergo an adaptive transformation. Specifically, agents $\{1, 4, 7\}$ form an Δ_3 group and move in directions different from agents $\{2, 3\}$ and $\{0, 5, 6\}$, who

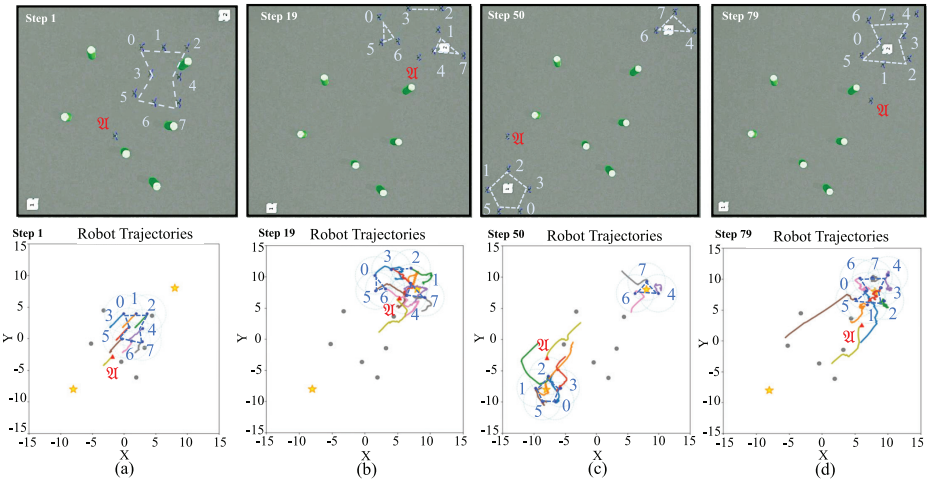


Fig. 10. Four typical cooperation scenarios in one episode of SITL. (a)–(d) Snapshots at high-level decision steps 1, 19, 50, and 79, respectively.

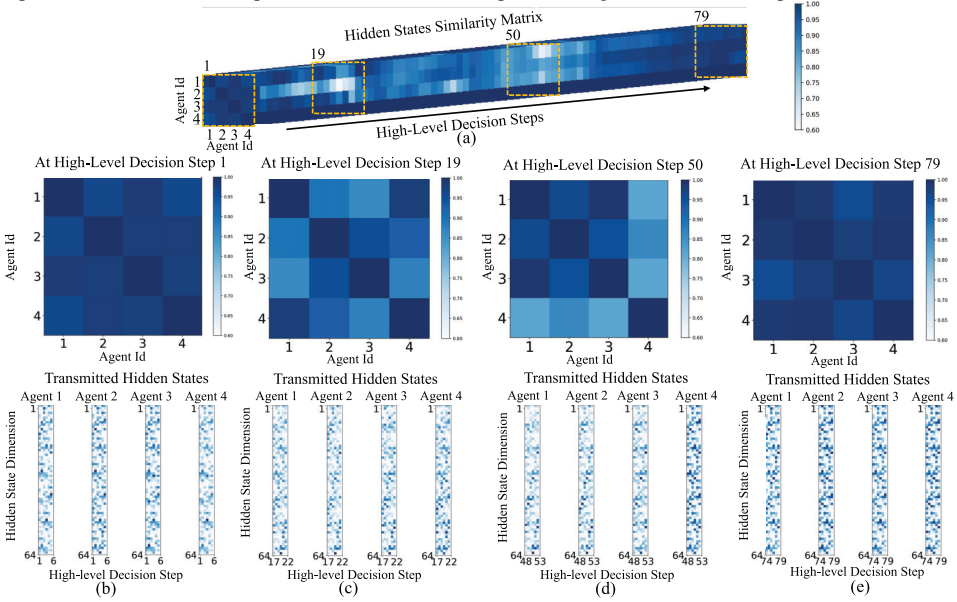


Fig. 11. Transmitted hidden states $\mathbf{m}$ for high-level decision steps and related similarity matrix analysis. (a) Hidden states similarity matrix variation for one episode. (b)–(e) Illustration of transmitted hidden states and similarity matrices at steps 1, 19, 50, and 79.

seem to be responsible for diverting the adversary's attention, according to the similar intention of $\{1, 4\}$ and $\{2, 3\}$ in Fig. 11(c). During the adversary approaches to the left side at step 50, agents $\{4, 6, 7\}$ reassess the safety of the target area $\mathbf{p}_2 = (8, 8)$ and automatically form Δ_3, while agents $\{0, 1, 2, 3, 5\}$ proceed to the anchor point $(-8, -8)$ and adaptively form a Δ_5 group, as shown in Fig. 10(c). Correspondingly, it can be observed from Fig. 11(d) that the communicated messages of agents $\{1, 2, 3\}$ maintain a similarity higher than 0.9, while the similarity with Agent 4 turns less than 0.85. Such an observation is consistent with the case where agents in two different groups choose different anchor points. Later at step 79, with the adversary moving close to $(-8, -8)$, agents $\{0, 1, 2, 3, 5\}$ escape from $\mathbf{p}_1$, reach the anchor point $(8, 8)$ and eventually form Δ_8 as shown in Fig. 10(d), when the messages become highly similar again as shown in Fig. 11(e).

c) Impact of real-world factors: To validate the robustness of our algorithm deployed at real-world UAV systems, we simulate more practical factors (e.g., winds and sensing deviation) with Gazebo plugins. Table VII records the times of critical events such as formation, navigation, and collision probability during 1000-step game in SITL. It shows that AT-M can competently handle environmental dynamics. For example, even confronted with strong wind, whose speed follows a Gaussian distribution with a mean of 8 m/s, the formation and navigation completion remain approximately 78.4% compared to that in a calm environment. Similarly, AT-M sustains 80% of its performance even when its input is skewed by a Gaussian distribution with a mean of 0.8 m. It is safe and reliable until the deviation is minor than the obstacle radius, which is 0.4 m in our case. Furthermore, to verify the effectiveness in a realistic distributed architecture, we deploy it across multiple devices in a LAN environment. For example, an NVIDIA Jetson TX2 NX is assigned to run a single onboard agent process, requiring 12.3 ms/decision, while a GeForce RTX 4090-equipped computer is responsible for seven other agents, consuming 0.96 ms/decision. Owing to the gap in computing efficiency, the average completion rate of asynchronous AT-M, denoted as SITL-M in Table VII, reaches 97.1% of the performance observed in simulation on a single device (denoted as SITL-S). As for CI-HRL, the average effective formation rate in 50 episodes of SITL-M is equivalent to 92.7% of that achieved by SITL-S.

TABLE VII
PERFORMANCE VALIDATION OF AT-M IN SITL WITH DIFFERENT SENSING DEVIATION, WIND SPEED, AND HARDWARE DEPLOYMENT

Critical Event	**F** (s)	**N** (s)	**C** (%)	**F** (s)	**N** (s)	**C** (%)
c		5			6	
SITL-S	693	722.4	0	711	645	0
Wind-3m/s	669.3	651.3	0	637	644	0
Wind-5m/s	643.2	579.2	0	586	632.5	0
Wind-8m/s	558	549	0	530	628	0
Deviation-0.3m	683	652	0	667	639	0
Deviation-0.5m	647.5	628.7	0	616	623	0.33
Deviation-0.8m	545.5	605.5	2.95	534	551	1.20
SITL-M	652.8	711.1	0	676.1	647.9	0
c		7			8	
SITL-S	746	707	0	712	696.5	0
Wind-3m/s	582	636	0	612.6	631.3	0
Wind-5m/s	577	585	0	593.7	615.7	0
Wind-8m/s	521	508	0	524	580	0
Deviation-0.3m	676.3	703.2	0	616.3	649	0
Deviation-0.5m	606	690.5	0.60	600.3	643	0.28
Deviation-0.8m	590	593	2.20	503	551	2.18
SITL-M	691.4	700.8	0	654	705.7	0

VI. CONCLUSION AND FUTURE WORK

In this work, toward accomplishing the CEFC task, we have proposed and validated a consensus inference-based hierarchical MARL framework (CI-HRL). Specifically, in low-level policy, we have implemented an AT-M to satisfy multiple coupled constraints and better balance the tradeoff between formation and navigation performance through dense random obstacles. Moreover, policy distillation has been adopted to achieve a more flexible adaptive formation. Meanwhile, in high-level control, to infer global information from the partially observed local state and implicitly establish the consensus, the ConsMAC methodology, has been designed in a novel supervised learning manner. Finally, we have conducted extensive experiments in MPE and SITL environments and successfully demonstrated the effectiveness and robustness of individual modules in CI-HRL. Moreover, the superiority of CI-HRL has been thoroughly validated.

Although CI-HRL has brilliant performance, there are still policy convergence challenges in super-large-scale scenarios. This limitation stems from the exponential growth of complexity for the centralized critic in the CTDE framework and the strict topological constraints in formation control. Potential solutions include fully decentralized training [51] or fractal-based hierarchical partitioning [52]. Meanwhile, we will carry out intense studies to further improve the stability of the MARL framework, optimize communication overhead, and develop methods to counter stronger adversaries. For example, to enhance UAV maneuverability in a 3-D environment, we will study the incorporation of an altitude planner for altitude adjustments while maintaining the swarm's fundamental 2-D behaviors. Furthermore, recent advances in adversarial RL [53], [54] provide promising directions for designing adaptive adversaries that can dynamically adjust their strategies during training. Integrating such methods could enhance the robustness of CI-HRL against sophisticated adversarial tactics.

APPENDIX A
DETAILS OF THE REWARD FUNCTION DESIGN

In this part, we introduce the details of the reward function in Section III-A2. For simplicity of representation, we denote the relative positions of agents in the group $k \in \{1, \ldots, \chi^{(t)}\}$ as $\mathbf{P}_k^{(t)} = \{\mathbf{p}_j^{(t)} - \bar{\mathbf{p}}_k^{(t)} | \forall j \in \mathcal{N}_k\}$ where $\bar{\mathbf{p}}_k^{(t)}$ is the center of the group and Δ_{n_k} is the target formation corresponding to k.

1) Formation Reward:

Toward implementing leader-free formation control and enhancing the robustness, we adopt the HD [42] to measure the formation error between the current and the expected formation. The HD between two topologies $\mathcal{E}_1$ and $\mathcal{E}_2$ is defined as $\mathrm{HD}(\mathcal{E}_1, \mathcal{E}_2) = \max_{\mathbf{x}\in\mathcal{E}_1} \min_{\mathbf{y}\in\mathcal{E}_2} \|\mathbf{x}-\mathbf{y}\|$. The formation reward for each group can be obtained as

$$R_f^{(t)} = -\sum_{k=1}^{\chi^{(t)}} \mathrm{HD}\left(\Delta_{n_k}, \mathbf{P}_k^{(t)}\right) - \omega_l R_f^{(t-1)} \tag{15}$$

where $R_f^{(0)} = 0$ and ω_l is the formation lag coefficient to better reflect the formation trend.

2) Navigation Reward:

The navigation reward simply uses an urgency-weighted Euclidean distance between agents and targets, which can be given by

$$R_{\mathrm{n}}^{(t)} = -\sum_{i\in\mathcal{N}} \sum_{\mathfrak{T}\in\mathcal{T}} \kappa_{\mathfrak{T}}^{(t)} \left\| \mathbf{p}_{i\to\mathfrak{T}}^{(t)} \right\|. \tag{16}$$

3) Task Accomplishment Reward:

The task accomplishment reward is awarded when each group k reaches the target area and maintains the formation and can be written as

$$R_t^{(t)} = \sum_{\mathfrak{T}\in\mathcal{T}} \kappa_{\mathfrak{T}}^{(t)} \sum_{k^*} \mathrm{TR}_{k^*\to\mathfrak{T}}^{(t)} \tag{17}$$

where

$$\mathrm{TR}_{k^*\to\mathfrak{T}}^{(t)} = \begin{cases} n_{k^*}, & \text{if } \left\| \bar{\mathbf{p}}_{k^*\to\mathfrak{T}}^{(t)} \right\| < \delta_{\mathrm{task}} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

with δ_{task} denoting the radius of the target area and $k^* \in \{1, \ldots, \chi^{(t)}\}$ indicates a group meeting a formation tolerance δ_{for}, that is, $\mathrm{HD}(\Delta_{n_{k^*}}, \mathbf{P}_{k^*}^{(t)}) < \delta_{\mathrm{for}}$. Besides, the urgency factor $\kappa_{\mathfrak{T}}^{(t)}$ will gradually decay when the agents stay in target area $\mathfrak{T}$ as follows:

$$\kappa_{\mathfrak{T}}^{(t+1)} = \max\left(\kappa_{\mathfrak{T}}^{(t)} - \omega_d \sum_{k^*} \mathrm{TR}_{k^*\to\mathfrak{T}}^{(t)}, 0\right) \tag{19}$$

where $\kappa_{\mathfrak{T}}^{(0)} = 1$ and ω_d is the decay factor.

4) Evasion Reward:

The evasion reward is designed to prevent the adversary and can be formulated as

$$R_e^{(t)} = -\sum_{i\in\mathcal{N}} \max\left(\delta_{a,e} - \left\| \mathbf{p}_{i\to\mathfrak{A}}^{(t)} \right\|, 0\right) \tag{20}$$

where $\delta_{a,e}$ is the alert distance of evasion.

TABLE VIII
PARAMETER SETTINGS OF THE REWARD

Parameters	Settings
Formation lag coefficient ω_l	0.3
Decay factor ω_d	0.003
Alert distance $\delta_{\mathrm{a,e}}$, $\delta_{\mathrm{a,c}}$	(2 m, 0.5 m)
Formation tolerance δ_{for}	1 m
The radius of the target area δ_{task}	3 m
Minimum safety distance δ_{s}	0.2 m
Collision constants $\omega_{cr_1}, \omega_{cr_2}, C_1, C_2$	$(24, 8, 3, 1)$

5) Collision Avoidance Reward:

The collision avoidance reward, which is designed to prevent obstacles, can be formulated as a summation of individual rewards, namely,

$$R_c^{(t)} = -\sum_{i\in\mathcal{N}}\sum_{j\neq i, j\in\mathcal{I}} \mathrm{CR}_{i\to j}^{(t)} \tag{21}$$

where $\mathcal{I}$ is the set of agents and obstacles, and for the minimum safety distance δ_s and collision alert distance $\delta_{a,c}$

$$\mathrm{CR}_{i\to j}^{(t)} = \begin{cases} \omega_{\mathrm{cr}_1}\left(\delta_s - d_{i,m}^{(t)}\right) + C_1, & d_{i,m}^{(t)} < \delta_s \\ \omega_{\mathrm{cr}_2}\left(\delta_{a,c} - d_{i,m}^{(t)}\right) + C_2, & \delta_s < d_{i,m}^{(t)} < \delta_{a,c} \\ 0, & d_{i,m}^{(t)} > \delta_a \end{cases} \tag{22}$$

with $d_{i,m}^{(t)} = \min(d_{i1}^{(t)}, \ldots, d_{iM}^{(t)})$ related to LiDAR and positive constants ω_{cr_1}, ω_{cr_2}, C_1, and C_2.

Furthermore, we have summarized the parameter settings of the reward in Table VIII.

REFERENCES

[1] M. Dhuheir, A. Erbad, and A. Al-Fuqaha, "Meta reinforcement learning for strategic IoT deployments coverage in disaster-response UAV swarms," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 6159–6164.

[2] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 6648–6653.

[3] P. Hu, Q. Pan, C. Zhao, and Y. Guo, "Transfer reinforcement learning for multi-agent pursuit-evasion differential game with obstacles in a continuous environment," *Asian J. Cont.*, vol. 26, no. 4, pp. 2125–2140, Feb. 2024.

[4] Z. M. Young and H. M. La, "Consensus, cooperative learning, and flocking for multiagent predator avoidance," *J. Adv. Robot. Syst.*, vol. 17, no. 5, pp. 1–19, Sep. 2020.

[5] V. Kedege, A. Czechowski, L. Stellingwerff, and F. Oliehoek, "Multi robot surveillance and planning in limited communication environments," in *Proc. 14th Int. Conf. Agents Artif. Intell.*, 2022, pp. 139–147.

[6] R. Konda, H. M. La, and J. Zhang, "Decentralized function approximated Q-learning in multi-robot systems for predator avoidance," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6342–6349, Oct. 2020.

[7] Z. Zhang, Q. Zong, D. Liu, X. Zhang, R. Zhang, and W. Wang, "A pursuit-evasion game on a real-city virtual simulation platform based on multi-agent reinforcement learning," in *Proc. Chin. Cont. Conf.*, Tianjin, China, Jul. 2023, pp. 6018–6023.

[8] Z. Yuan, C. Yao, X. Liu, Z. Gao, and W. Zhang, "Multiagent formation control and dynamic obstacle avoidance based on deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 21, no. 6, pp. 4672–4682, Jan. 2025.

[9] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–16.

[10] C. Yan et al., "Collision-avoiding flocking with multiple fixed-wing UAVs in obstacle-cluttered environments: A task-specific curriculum-based MADRL approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 10894–10908, Jun. 2023.

[11] J. Chai, Y. Zhu, and D. Zhao, "NVIF: Neighboring variational information flow for cooperative large-scale multiagent reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17829–17841, Dec. 2024.

[12] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 7234–7284, 2020.

[13] C. Guan et al., "Efficient multi-agent communication via self-supervised information aggregation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Nov. 2022, pp. 1–13.

[14] J. Liu et al., "Maximum entropy heterogeneous-agent reinforcement learning," in *Proc. Int. Conf. Learn. Repre.*, Vienna, Austria, May 2024, pp. 1–12.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[16] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Nov. 2022, pp. 24611–24624.

[17] Y. Xie et al., "Multi-UAV behavior-based formation with static and dynamic obstacles avoidance via reinforcement learning," 2024, *arXiv:2410.18495*.

[18] Y. Yan et al., "Relative distributed formation and obstacle avoidance with multi-agent reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Philadelphia, PA, USA, May 2022, pp. 1661–1667.

[19] Y. Jin, S. Wei, J. Yuan, and X. Zhang, "Hierarchical and stable multiagent reinforcement learning for cooperative navigation control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 90–103, Jan. 2023.

[20] Z. Lu, R. Li, M. Lei, C. Wang, Z. Zhao, and H. Zhang, "Self-critical alternate learning based semantic broadcast communication," 2023, *arXiv:2312.01423*.

[21] C. Zhu, M. Dastani, and S. Wang, "A survey of multi-agent deep reinforcement learning with communication," in *Proc. 23rd Int. Conf. Auton. Agents Multiagent Syst.*, vol. 38, Auckland, New Zealand, Jan. 2024, pp. 2845–2847.

[22] A. Amirkhani and A. H. Barshooi, "Consensus in multi-agent systems: A review," *Artif. Intell. Rev.*, vol. 55, no. 5, pp. 3897–3935, 2022.

[23] X. Yang, H. Zhang, and Z. Wang, "Data-based optimal consensus control for multiagent systems with policy gradient reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3872–3883, Aug. 2022.

[24] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Dec. 2016, pp. 2252–2260.

[25] A. Das et al., "TarMAC: Targeted multi-agent communication," *Proc. 36th Int. Conf. Mach. Learn.*, Sacramento, CA, USA, pp. 1538–1546, May 2019.

[26] W. Kim, J. Park, and Y. Sung, "Communication in multi-agent reinforcement learning: Intention sharing," in *Proc. Int. Conf. Learn. Repre.*, May 2021, pp. 1–15.

[27] Y. Wang, F. Zhong, J. Xu, and Y. Wang, "ToM2C: Target-oriented multi-agent communication and cooperation with theory of mind," in *Proc. Int. Conf. Learn. Repre.*, Jan. 2022, pp. 1–17.

[28] Z. Xu et al., "Consensus learning for cooperative multi-agent reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, Washington, DC, USA, Jun. 2023, vol. 37, no. 10, pp. 11726–11734.

[29] S. Pateria, B. Subagdja, A.-H. Tan, and C. Quek, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Comput. Surveys*, vol. 54, no. 5, pp. 1–35, 2021.

[30] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–17, Jul. 2022.

[31] G. Xu, W. Jiang, Z. Wang, and Y. Wang, "Autonomous obstacle avoidance and target tracking of UAV based on deep reinforcement learning," *J. Intell. Robotic Syst.*, vol. 104, no. 4, pp. 60–79, Apr. 2022.

[32] R. Zhang, Q. Zong, X. Zhang, L. Dou, and B. Tian, "Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7900–7909, Oct. 2023.

[33] H. Yang, P. Ge, J. Cao, Y. Yang, and Y. Liu, "Large scale pursuit-evasion under collision avoidance using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Detroit, MI, USA, Oct. 2023, pp. 2232–2239.

[34] Z. Deng and Z. Kong, "Multi-agent cooperative pursuit-defense strategy against one single attacker," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5772–5778, Oct. 2020.

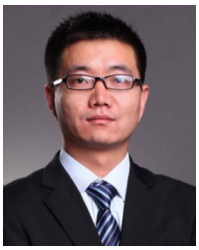
[35] P. Dayan and G. E. Hinton, "Feudal reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 5, 1992, pp. 1–8.
[36] T. G. Dietterich, "Hierarchical reinforcement learning with the MAXQ value function decomposition," *J. Artif. Intell. Res.*, vol. 13, pp. 227–303, Nov. 2000.
[37] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 1726–1734.
[38] A. Levy, G. Konidaris, R. Platt, and K. Saenko, "Learning multi-level hierarchies with hindsight," 2017, *arXiv:1712.00948*.
[39] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 6235–6240.
[40] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," M.Sc. thesis, Dept. Autom. Control, Lund Univ., Lund, Sweden, 2008.
[41] P. Yan, J. Guo, X. Su, and C. Bai, "Long-term tracking of evasive urban target based on intention inference and deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16886–16900, Nov. 2024.
[42] C. Pan, Y. Yan, Z. Zhang, and Y. Shen, "Flexible formation control using Hausdorff distance: A multi-agent reinforcement learning approach," in *Proc. Euro. Sign. Conf.*, Glasgow, U.K., Aug. 2022, pp. 972–976.
[43] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. Int. Conf. Learn. Repre.*, May 2016, pp. 1–17.
[44] A. A. Rusu et al., "Policy distillation," in *Proc. Int. Conf. Learn. Represent.*, San Juan, Puerto Rico, May 2016, pp. 1–13.
[45] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.
[46] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sendai, Japan, vol. 3, Oct. 2004, pp. 2149–2154.
[47] Y. Wang, S. Huang, Y. Yu, C. Li, P. A. Hoeher, and A. C. K. Soong, "Recent progress on 3GPP 5G positioning," in *Proc. IEEE 97th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2023, pp. 1–6.
[48] A. Vadduri, A. Benjwal, A. Pai, E. Quadros, A. Kammar, and P. Uday, "Precise payload delivery via unmanned aerial vehicles: An approach using object detection algorithms," 2023, *arXiv:2310.06329*.
[49] Z. Sui, Z. Pu, J. Yi, and S. Wu, "Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2358–2372, Jun. 2021.
[50] K. Xiao, S. Tan, G. Wang, X. An, X. Wang, and X. Wang, "XTDrone: A customizable multi-rotor UAVs simulation platform," in *Proc. 4th Int. Conf. Robot. Autom. Sci. (ICRAS)*, Jun. 2020, pp. 55–61.
[51] C. Ma, A. Li, Y. Du, H. Dong, and Y. Yang, "Efficient and scalable reinforcement learning for large-scale network control," *Nature Mach. Intell.*, vol. 6, no. 9, pp. 1006–1020, 2024.
[52] J. Wu, D. Li, Y. Yu, L. Gao, J. Wu, and G. Han, "An attention mechanism and adaptive accuracy triple-dependent MADDPG formation control method for hybrid UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 11648–11663, Sep. 2024.
[53] L. Schott et al., "Robust deep reinforcement learning through adversarial attacks and training: A survey," 2024, *arXiv:2403.00420*.
[54] L. Yuan et al., "Robust multi-agent coordination via evolutionary generation of auxiliary adversarial attackers," in *Proc. AAAI Conf. Artif. Intell.*, Washington, DC, USA, Jan. 2023, pp. 11753–11762.
[55] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 1999, pp. 1–7.
[56] Y. Hu et al., "Learning to utilize shaping rewards: A new approach of reward shaping," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 15931–15941.

Yuming Xiang received the M.S. degree from the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, in March 2025

His research interests include multiagent reinforcement learning.

Sizhao Li received the M.S. degree from the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, in March 2025

His research interests include multiagent reinforcement learning and large language models.



Rongpeng Li (Senior Member, IEEE) is currently an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He was a Research Engineer with the Wireless Communication Laboratory, Huawei Technologies Company Ltd., Shanghai, China, from August 2015 to September 2016. He was a Visiting Scholar with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K., from February 2020 to August 2020. His research interests include networked intelligence for communications evolving (NICE).

Dr. Li received the Wu Wenjun Artificial Intelligence Excellent Youth Award in 2021 and the sponsorship "of the National Postdoctoral Program for Innovative Talents" in 2016, respectively. He serves as an Editor for *China Communications*.

Zhifeng Zhao (Senior Member, IEEE) received the B.E. degree in computer science, the M.E. and Ph.D. degrees in communication and information systems from the PLA University of Science and Technology, Nanjing, China, in 1996, 1999, and 2002, respectively.

From 2002 to 2004, he acted as a Post-Doctoral Researcher with Zhejiang University, Hangzhou, China, where his researches were focused on multimedia next-generation networks (NGNs) and soft switch technology for energy efficiency. From 2005 to 2006, he acted as a Senior Researcher with the PLA University of Science and Technology, where he performed research and development on advanced energy-efficient wireless routers, ad hoc network simulators, and cognitive mesh networking testbeds. From 2006 to 2019, he was an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University. He is currently with Zhejiang Lab, Hangzhou, as the Chief Engineering Officer. His research interests include software-defined networks (SDNs), wireless networks in 6G, computing networks, and collective intelligence.

Dr. Zhao is the Symposium Co-Chair of ChinaCom 2009 and 2010. He is the Technical Program Committee (TPC) Co-Chair of the 10th IEEE International Symposium on Communication and Information Technology (ISCIT 2010).

Honggang Zhang (Fellow, IEEE) was a Professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He was an Honorary Visiting Professor with the University of York, York, U.K., and an International Chair Professor of Excellence with the Université Européenne de Bretagne, Supélec, France. He is a Professor with the School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China. His research interests include cognitive radio networks, semantic communications, green communications, machine learning, artificial intelligence, intelligent computing, and Internet of Intelligence.

Dr. Zhang is a co-recipient of the 2021 IEEE Communications Society Outstanding Paper Award and the 2021 IEEE INTERNET OF THINGS JOURNAL (IOT-J) Best Paper Award. He served as the Chair of the Technical Committee on Cognitive Networks of the IEEE Communications Society from 2011 to 2012. He was the founding Chief Managing Editor of *Intelligent Computing*, a Science Partner Journal. He was the leading Guest Editor for the Special Issues on Green Communications of the *IEEE Communications Magazine*. He served as a Series Editor for the *IEEE Communications Magazine* (Green Communications and Computing Networks Series) from 2015 to 2018. He is the Associate Editor-in-Chief of *China Communications*.