AT-Drone: Benchmarking Adaptive Teaming in Multi-Drone Pursuit

Yang Li[1], **Junfan Chen**[2], **Feng Xue**[3], **Jiabin Qiu**[4], **Wenbin Li**[4], **Qingrui Zhang**[3], **Ying Wen**[2] *, **Wei Pan**[1] *
[1] University of Manchester; [2] Shanghai Jiao Tong University;
[3] Sun Yat-sen University; [4] Nanjing University;

Abstract: Adaptive teaming—the capability of agents to effectively collaborate with unfamiliar teammates without prior coordination—is widely explored in virtual video games but overlooked in real-world multi-robot contexts. Yet, such adaptive collaboration is crucial for real-world applications, including border surveillance, search-and-rescue, and counter-terrorism operations. Such real-world scenarios involve dynamic environments, unpredictable target behaviors, and continuously changing team compositions, demanding exceptional adaptiveness. These challenges highlight the limitations of pre-coordinated strategies, which cannot adequately support seamless collaboration under uncertainty. To address this gap, we introduce AT-Drone, the first dedicated benchmark explicitly designed to facilitate comprehensive training and evaluation of adaptive teaming strategies in multi-drone pursuit scenarios. AT-Drone makes the following key contributions: (1) An adaptable AT-Drone simulator, which provides an adaptable simulation environment configurator for intuitive and rapid setup of adaptive teaming multi-drone pursuit tasks, including four predefined pursuit environments. (2) A streamlined real-world deployment pipeline that seamlessly translates simulation insights into practical drone evaluations using edge devices and Crazyflie drones. (3) A novel algorithm zoo integrated with a distributed training framework, featuring diverse algorithms explicitly tailored, for the first time, to multi-pursuer and multi-evader settings. (4) Standardized evaluation protocols with newly designed unseen drone zoos, explicitly designed to rigorously assess the performance of adaptive teaming. Comprehensive experimental evaluations across four progressively challenging multi-drone pursuit scenarios confirm AT-Drone's effectiveness in advancing adaptive teaming research. Real-world drone experiments further validate its practical feasibility and utility for realistic robotic operations.

Keywords: adaptive teaming,multi-robot collaboration,multi-drone pursuit

1 Introduction

Multi-drone collaboration has become increasingly critical for a variety of real-world applications, including disaster response, border surveillance, and search-and-rescue missions [1, 2, 3]. The success of these missions heavily relies on drones' capability to effectively coordinate in real-time within dynamic environments with changing team compositions and unpredictable target behaviors. For example, in disaster scenarios, drones may become damaged or depleted during operation, necessitating rapid integration of backup drones to sustain mission effectiveness. Adaptive teaming directly addresses these challenges by enabling drones to dynamically collaborate with previously unseen teammates, significantly enhancing the operational robustness and flexibility of drone fleets.

However, current multi-drone collaboration methodologies predominantly rely on pre-defined coordination mechanisms or extensive prior interactions among drones, thus limiting their adaptability to

*Corresponding authors.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

Table 1: Comparison of related work. Grey rows indicate multi-drone pursuit literature, pink rows highlight adaptive teaming studies. "AT w/o TM" and "AT w/ TM" denote adaptive teaming without and with teammate modeling.

Related Work	Problem Setting				Task		Method	
	# Learner	# Unseen	# Opponent	Action Space	Main Related Task	Real-world?	AT w/o TM?	AT w/ TM?
Voronoi Partitions [6]	Multi	0	1	Continuous	Pursuit–evasion Game	No	No	No
Bio-pursuit [5]	Multi	0	Multi	Continuous	Prey–predator Game	No	No	No
Uncertainty-pursuit [4]	Multi	0	1	Continuous	Pursuit–evasion Game	No	No	No
M3DDPG [10]	Multi	0	1	Continuous	Prey–predator Game	No	No	No
Pursuit-TD3[9]	Multi	0	1	Continuous	**Multi-drone Pursuit**	**Yes**	No	No
DACOOP-A[2]	Multi	0	1	Discrete	**Multi-drone Pursuit**	**Yes**	No	No
GM-TD3 [23]	Multi	0	1	Continuous	Prey–predator Game	No	No	No
DualCL [7]	Multi	0	1	Continuous	**Multi-drone Pursuit**	No	No	No
HOLA-Drone [24]	1	Multi	Multi	Continuous	**Multi-drone Pursuit**	**Yes**	**Yes**	No
Other-play [12]	1	1	0	Discrete	Lever Game; Hanabi	No	**Yes**	No
Overcooked-AI [25]	1	1	0	Discrete	Overcooked	No	**Yes**	No
TrajDi [26]	1	1	0	Discrete	Overcooked	No	**Yes**	No
MEP [27]	1	1	0	Discrete	Overcooked	No	**Yes**	No
LIPO [28]	1	1	0	Discrete	Overcooked	No	**Yes**	No
HSP [29]	1	1	0	Discrete	Overcooked	No	**Yes**	No
COLE [30]	1	1	0	Discrete	Overcooked	No	**Yes**	No
ZSC-Eval [15]	1	1	0	Discrete	Overcooked	No	**Yes**	No
PLASTIC [31]	1	Multi	Multi	Discrete	Prey-predator Game	No	No	**Yes**
AATeam [32]	1	1	2	Discrete	Half Field Offense	No	No	**Yes**
LIAM [20]	1	Multi	Multi	Discrete	LBF; Prey-predator Game	No	No	**Yes**
GPL [33]	1	Multi	Multi	Discrete	LBF; Wolfpack; FortAttack	No	No	**Yes**
CIAO [34]	1	Multi	Multi	Discrete	LBF; Wolfpack	No	No	**Yes**
NAHT [21]	Multi	Multi	Multi	Discrete	StarCraft; MPE	No	No	**Yes**

unexpected or new teammates. Traditional optimization-based approaches [4, 5, 6] and reinforcement learning methods [2, 7, 8, 9, 10, 11] typically utilize fixed conventions, roles, or communication protocols, hindering their performance when encountering unfamiliar teammates or environments.

Conversely, existing adaptive teaming research—such as zero-shot coordination (ZSC) [12] and ad-hoc teamwork (AHT) [13]—mainly focuses on simulated environments with discrete action spaces, exemplified by video games such as Overcooked [14, 15], Hanabi [12, 16, 17, 18], and Predator-Prey [19, 20]. Recent advancements like NAHT [21] extend these paradigms to multiple learners but remain restricted within discrete-action domains such as the SMAC [22], limiting their real-world applicability.

As summarized in Table 1, there is a clear lack of benchmarks tailored specifically for studying adaptive teaming in complex, real-world scenarios involving multi-drone collaboration. To address this critical gap, we introduce AT-Drone, the first unified benchmark explicitly designed to integrate adaptive teaming methods from machine learning into practical multi-drone robotics applications. AT-Drone provides a comprehensive and standardized training and evaluation framework, facilitating rapid assessment of adaptive teaming algorithms and effectively bridging theoretical innovations with real-world robotic deployments.

AT-Drone consists of four main components designed to thoroughly study adaptive teaming in multi-drone pursuit tasks: **1. A customizable simulation environment:** AT-Drone includes four progressively challenging multi-drone pursuit environments, systematically varying in obstacle complexity, evader numbers, and task difficulty, enabling rigorous testing of adaptive strategies across diverse operational conditions. **2. Streamlined real-world deployment pipeline:** AT-Drone integrates practical deployment pipelines that leverage motion capture systems and edge devices (such as Nvidia Jetson Orin Nano). Real-world experiments with Crazyflie drones validate the benchmark's fidelity and reflect its potential for advancing more complex real-world applications in the future. **3. A novel algorithm zoo:** AT-Drone introduces a distributed training infrastructure comprising seven adaptive teaming algorithms adapted and extended from discrete video game environments. To the best of our knowledge, this benchmark represents the first systematic exploration of adaptive teaming strategies tailored to multi-pursuit multi-evader drone pursuit scenarios. **4. Standardized evaluation protocols:** AT-Drone provides three distinct "unseen drone zoo" configurations, each demanding unique adaptive collaboration strategies, alongside four specialized evaluation metrics designed to systematically assess algorithm adaptability and robustness.

2 Related Work

As summarised in Table 1, we provide a detailed comparison of related methods across key dimensions, including problem formulation, task scope, and methodological approaches, highlighting the unique positioning of our benchmark within the literature.

Multi-agent pursuit-evasion. Multi-agent pursuit-evasion is closely related to the multi-drone pursuit task. Most existing methods rely on pre-coordinated strategies specifically designed for particular pursuit-evasion scenarios. Traditional approaches often rely on heuristic [5] or optimisation-based strategies [6, 4]. In recent years, deep reinforcement learning (DRL) has been widely adopted for pre-coordinated multi-drone pursuit tasks. M3DDPG [10] and GM-TD3 [23] extend standard DRL algorithms, such as TD3 [35] and DDPG [36], specifically for multi-agent pursuit in simulated environments. Pursuit-TD3 [9] applies the TD3 algorithm to pursue a target with multiple homogeneous agents, which is validated through both simulations and real-world drone demonstrations. Zhang et al. [2] introduces DACOOP-A, a cooperative pursuit algorithm that enhances reinforcement learning with artificial potential fields and attention mechanisms, which is evaluated in real-world drone systems. DualCL [7] addresses multi-UAV pursuit-evasion in diverse environments and demonstrates zero-shot transfer capabilities to unseen scenarios, though only in simulation. The most recent work, HOLA-Drone [24], claims to be the first ZSC framework for multi-drone pursuit. However, it is limited to controlling a single learner, restricting its applicability to broader multi-agent settings.

Adaptive Teaming. The adaptive teaming paradigm can be broadly categorised into two aspects: adaptive teaming without teammate modelling (AT w/o TM) and adaptive teaming with teammate modelling (AT w/ TM), which correspond to the zero-shot coordination (ZSC) and ad-hoc teamwork (AHT) problems in the machine learning community, respectively. AT w/o TM focuses on enabling agents to coordinate with unseen teammates without explicitly modelling their behaviours. Other-Play [12] introduces an approach that leverages symmetries in the environment to train robust coordination policies, applied to discrete-action tasks like the Lever Game and Hanabi. Similarly, methods such as Overcooked-AI [25], TrajDi [26], MEP [27], LIPO [28], and ZSC-Eval [15] study collaborative behaviours in Overcooked, where agents learn generalisable coordination strategies with diverse unseen partners. While these approaches demonstrate promising results, they are limited to single-learner frameworks in simplified, discrete-action domains like Overcooked and Hanabi. They lack scalability to multi-agent settings, continuous action spaces, and the complexities of real-world applications. AT w/ TM, on the other hand, explicitly models the behaviour of unseen teammates to facilitate effective collaboration. Early methods like PLASTIC [31] reuse knowledge from previous teammates or expert input to adapt to new teammates efficiently. AaTeam [32] introduces attention-based neural networks to dynamically process and respond to teammates' behaviours in real-time. More advanced approaches, such as LIAM [20], employ encoder-decoder architectures to model teammates using local information from the controlled agent. GPL [33] and CIAO [34] leverage GNNs to address the challenges of dynamic team sizes in AHT. Extending from the AHT settings, NAHT [21] enables multiple learners to collaborate and interact with diverse unseen partners in N-agent scenarios. Despite their progress, these methods remain confined to discrete action spaces and simulated benchmarks, limiting their applicability to real-world, continuous-action tasks.

3 The Benchmark: AT-Drone

3.1 Problem Formulation

Definition 3.1 (Adaptive Teaming in Multi-Drone Pursuit)**.** Adaptive teaming in multi-drone pursuit involves training a set of $N \in \{1, 2, \dots\}$ drone agents, referred to as learners, to dynamically coordinate with $M \in \{1, 2, \dots\}$ previously unseen partners. The objective is to pursue $K \in \{1, 2, \dots\}$ targets without collisions, optimizing the overall return.

Let $\mathcal{C}$ represent the cooperative team, comprising N learners and M uncontrolled teammates. The set of uncontrolled teammates is denoted by $\mathcal{U}$. In the multi-drone pursuit task, there exists a set of opponents, denoted as $\mathcal{E}$. Adaptive teaming can be effectively modeled as an extended **A**daptive **T**eaming **Dec**entralized **P**artially **O**bservable **M**arkov **D**ecision **P**rocess (AT-Dec-POMDP). AT-Dec-POMDP is defined by the tuple $(\mathcal{S}, \mathcal{C}, \mathcal{A}, \mathcal{P}, r, \mathcal{O}, \gamma, T)$, where: where $\mathcal{S}$ is the joint state space; $\mathcal{C}$ denotes the set of cooperative agents, consisting of learners ($\mathcal{N}$) and uncontrolled teammates ($\mathcal{M}$), where $\mathcal{M}$ is sampled according to $\mathcal{P}_{\text{u}}(\mathcal{M}|\mathcal{U})$ from the complete set of uncontrolled teammates $\mathcal{U}$; $\mathcal{A} = \times_{j=1}^{C} \mathcal{A}^j$ is the joint action space, where $C = N + M$ is the team size; $\mathcal{P}(s'|s, a)$ is the

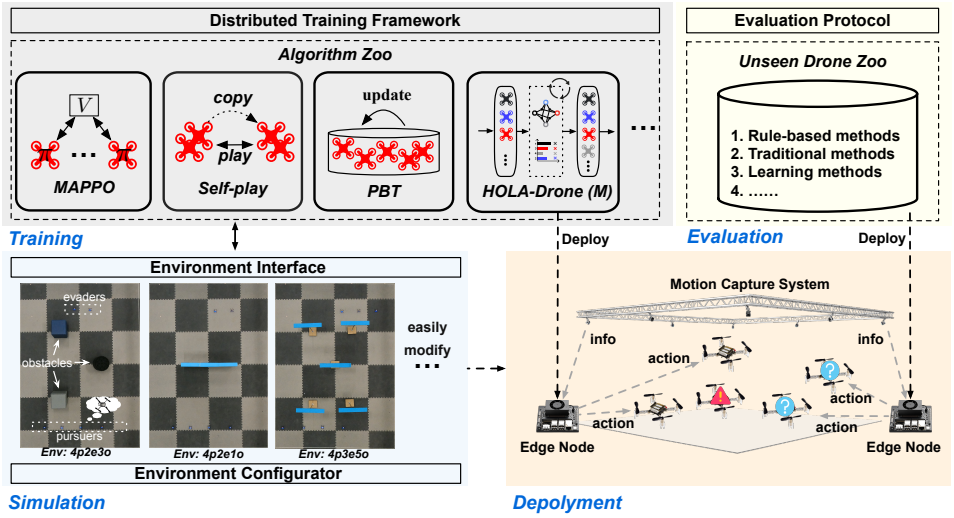

Figure 1: Overview of the AT-Drone Benchmark, comprising four key components: (I) a customizable simulation environment featuring varied multi-drone pursuit tasks with adjustable complexity; (II) a streamlined real-world deployment pipeline employing motion capture systems and edge devices to facilitate realistic drone validation; (III) a distributed training framework equipped with diverse adaptive teaming algorithms for multi-drone pursuit task; and (IV) standardized evaluation protocols, leveraging diverse unseen teammate configurations to rigorously evaluate adaptive teaming performance and robustness across distinct strategies.

transition probability function, representing the probability of transitioning to state $s' \in \mathcal{S}$ given the current state $s \in \mathcal{S}$ and joint action $a \in \mathcal{A}$; $r(s, a)$ is the reward function, representing the team's reward in state s after taking action a; $\mathcal{O}$ is the joint observation space, with $\mathcal{O}(o|s)$ describing the probability of generating observation o given state s; $\gamma \in [0, 1]$ is the discount factor; and T is the task horizon.

Additionally, we denote the policy of agent j as π^j, through which the agent selects an action $a_t^j \in \mathcal{A}^j$, and the policies of the N learners (π^i, for $i \in \mathcal{N}$) are learnable; we consider two approaches for defining these policies: with and without teammate modeling. Adaptive teaming without teammate modeling is closely related to the zero-shot coordination problem [12, 25], where learners could coordinate with unseen teammates. Specifically, the policy for a learner i is represented as $\pi^i(a_t^i \mid \tau_t^i)$, where τ_t^i denotes the learner's observation history up to time t. On the other hand, adaptive teaming with teammate modeling aligns closely with the ad-hoc teamwork paradigm [13], where agents could coordinate with previously unknown teammates by explicitly modeling their behavior and characteristics. In this case, the policy is defined as $\pi^i(a_t^i \mid \tau_t^i, f(\tau_t^i))$. The joint action $a_t = (a_t^1, \ldots, a_t^C)$ determines the next state $s_{t+1} \sim \mathcal{P}(s_{t+1} \mid s_t, a_t)$, and all agents receive a shared reward $r(s_t, a_t)$. The goal of adaptive teaming is to learn policies $\{\pi^i\}_{i \in \mathcal{N}}$ that maximize the expected discounted return: $\mathcal{J} = \mathbb{E}[R(\tau)] = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)\right]$, where τ denotes the trajectory.

3.2 Simulation and Deployment

Simulation. The simulation module provides a highly customizable and intuitive framework through the environment configurator and the Gymnasium-based environment interface, enabling efficient setup and execution of multi-drone pursuit scenarios. The environment configurator organizes simulation parameters into three distinct categories for easy customization: players, site, and task (see Fig. 5 in Appendix A). The players category specifies the numbers, velocities, and characteristics of learners, unseen teammates, and evaders, and optionally incorporates an unseen drone zoo to simulate varied adaptive teaming conditions. The site category allows users to adjust the simulation environment's physical properties, including map dimensions and obstacle configurations, enabling a

wide range of scenario complexities. The task category defines rules and objectives unique to each pursuit scenario, determining conditions for success and interaction dynamics.

Leveraging the configurator, we systematically design four progressive multi-drone pursuit environments in the benchmark, named `4p2e3o`, `4p2e1o`, `4p2e5o`, and `4p3e5o`, indicating the number of pursuers (p), evaders (e), and obstacles (o). Screenshots of these real-world environments are provided in Fig. 4 in the Appendix. Evaders spawn within a specified region measuring 3.2m wide and 0.6m high, with pursuers spawning in a similar area. Obstacle configurations vary significantly across environments to introduce different complexity levels. Environment `4p2e3o`, considered easy, contains three distributed obstacles (two cubes and one cylinder), providing ample pursuit space. Although environment `4p2e1o` has only a single central obstacle, it poses a slightly higher difficulty due to evaders having greater freedom of movement. Environments featuring five obstacles (`4p2e5o` and `4p3e5o`) are challenging, requiring sophisticated maneuvering due to densely packed obstacles restricting drone movements. Specifically, `4p2e5o` is categorized as hard, while `4p3e5o` is identified as the most challenging scenario (superhard), necessitating advanced coordination strategies for successful adaptive teaming.

In these environments, each agent's observation includes: (1) the relative position and bearing to each evader within its perception range, (2) the distance and angle to the nearest obstacle, and (3) the relative position and bearing of nearby teammates. Observations are structured as normalized continuous vectors, with masking applied to represent occluded entities clearly. The action space is continuous, defined within the range $[-1, 1]$, directly corresponding to angular steering adjustments that control the drone's directional rotation during pursuit. The reward function incentivizes efficient pursuit behaviors through positive rewards for capturing evaders and additional shaped rewards proportional to the agents' proximity improvement towards targets. Safety is enforced via proximity-based penalties for approaching too close to obstacles or teammates, effectively discouraging risky behaviors and promoting cooperative, safe navigation.

Deployment. As shown on the right side of Fig. 1, the AT-Drone benchmark supports real-world deployment within a 3.6m$\times$ 5m area by seamlessly integrating edge computing nodes—such as the Nvidia Jetson Orin Nano and personal laptops—with Crazyflie drones. Specifically, we utilize the FZMotion system to perform real-time position tracking, transmitting positional data in point cloud format to Crazyswarm, where it is processed and fed into the decision-making policies. Policies for both the adaptive learners and unseen drone partners (sampled from the unseen drone pool) run on edge computing nodes that serve as inference engines. The Crazyswarm platform and the adaptive teaming policies are deployed separately across two edge devices: a Lenovo ThinkPad T590 laptop and a Jetson Orin Nano. These nodes handle the reception of drone position data from the motion capture system, execute the adaptive teaming algorithms, and transmit control commands to the Crazyflie drones via Crazyradio PA. Upon receiving control signals, the Crazyflie drones execute the maneuvers using their onboard Mellinger controller, ensuring accurate and responsive trajectory tracking. Overall, this real-world deployment setup allows us to directly evaluate the applicability of learned policies on physical drone systems, effectively bridging the gap between simulation and real-world application.

3.3 Training and Evaluation

Training. As illustrated in Fig. 1, the AT-Drone framework employs a distributed training architecture leveraging multiple parallel environments to efficiently scale the learning processes. To the best of our knowledge, this is the first study specifically targeting adaptive teaming strategies in multi-pursuer multi-evader drone pursuit scenarios, highlighting the novelty and importance of establishing a comprehensive algorithm zoo for rigorous benchmarking and broader community adoption.

To further boost zero-shot coordination capabilities, AT-Drone employs self-play and PBT strategies. Self-play enables drones to iteratively refine policies via competitive interactions against progressively updated versions of themselves, significantly improving their adaptability and coordination. Simultaneously, PBT [25] facilitates extensive exploration and efficient knowledge sharing among diverse model populations, thus enhancing generalization to dynamic pursuit tasks. Both self-play

and PBT are implemented using Independent PPO (IPPO) [37], providing a robust, scalable training environment ideal for real-world multi-drone collaboration.

Recently, the HOLA-Drone method [24] introduces a hypergraphical-form game to model single-learner scenarios involving interactions with multiple unseen teammates. However, this method faces significant limitations when extended to more complex multi-learner scenarios common in multi-drone pursuit tasks. To overcome these challenges, we propose **HOLA-Drone V2**, an enhanced and generalized approach. Our key improvements include the construction of a preference hypergraph to explicitly identify and retain optimal teammate interactions, combined with a novel max-min preference oracle. This oracle systematically identifies challenging teammate subsets and iteratively optimizes drone strategies to robustly handle these scenarios. By dynamically adjusting the learner strategy set, our method significantly improves adaptability and coordination effectiveness in realistic multi-agent environments. Comprehensive algorithmic details, formal definitions, and implementations are provided in Appendix C.

In addition to zero-shot coordination methods, we introduce an ad-hoc teamwork algorithm, **NAHT-D** (NAHT for Drones), based on the recent NAHT framework [21]. Unlike the original NAHT method for discrete-action tasks (e.g., SMAC), NAHT-D is adapted for continuous-action drone scenarios, crucial for realistic maneuvering and pursuit tasks. NAHT-D efficiently models unseen drone teammates by integrating a specialized teammate-modeling network into MAPPO [38]. The teammate-modeling network employs an autoencoder, taking the past k steps of observations and actions to reconstruct teammates' current action distributions. For continuous action spaces, we use KL divergence as the reconstruction loss. The encoder's output embedding captures teammate behaviors and is combined with the agent's current observations as input to the policy network. For detailed algorithmic implementation, please refer to Appendix D.

Evaluation. To rigorously evaluate adaptive teaming strategies within multi-drone pursuit scenarios, we design a structured evaluation protocol consisting of multiple distinct unseen drone zoos and clearly defined evaluation metrics.

We construct a set of **unseen drone zoos** to introduce diverse and challenging teammate behaviors, spanning rule-based, bio-inspired, and learning-based methods. Specifically, these zoos include: (1) the Greedy Drone, employing a straightforward pursuit strategy focused on the nearest evader while dynamically avoiding collisions; (2) the VICSEK Drone, inspired by swarm behaviors, optimizing collective drone movement for pursuit and collision avoidance; and (3) the Self-Play Drones, trained using randomized IPPO-based self-play to generate diverse and unpredictable coordination behaviors. To ensure thorough evaluation, we define three distinct configurations of unseen drone partners: **Unseen Zoo 1**: Greedy drones only, highlighting direct pursuit behaviors. **Unseen Zoo 2**: Includes two IPPO self-play drone policies demonstrating different coordination skill levels—one highly coordinated (70% success) and one less coordinated (54% success)—introducing variability in teaming performance. **Unseen Zoo 3**: Combines all drones from the previous zoos, randomly selecting partners each episode to maximize behavioral diversity and unpredictability. Detailed descriptions and implementations of these drone behaviors are provided in Appendix B.

We adopt four quantitative metrics to systematically evaluate adaptive teaming performance: **Success Rate (SUC)**: Percentage of episodes successfully completed, defined by capturing both evaders within 0.2 meters. **Collision Rate (COL)**: Frequency of collisions between drones (threshold of 0.2 meters) or drones and obstacles (threshold of 0.1 meters), assessing operational safety. **Average Success Timesteps (AST)**: Mean number of timesteps required for successful task completion, indicating pursuit efficiency. **Average Reward (REW)**: Overall quality and efficiency of agent performance across episodes.

4 Experiment

In this section, we assess baseline methods from our algorithm zoo to validate their practical effectiveness in multi-drone pursuit tasks across four progressively challenging environments: 4p2e3o, 4p2e1o, 4p2e5o, and 4p3e5o. The experiments are organized into two primary components: (1)

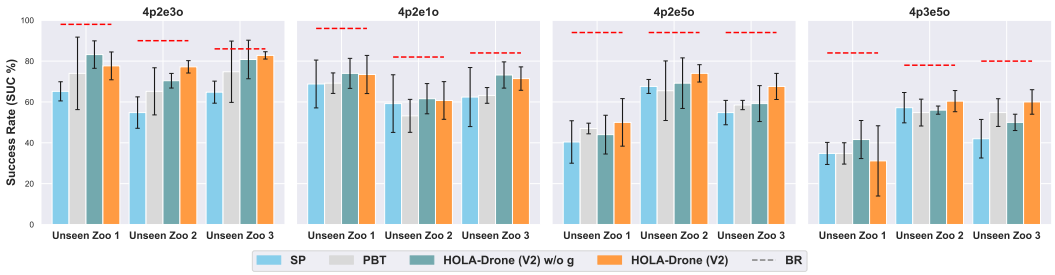

Figure 2: Success rate (SUC) across different difficulty levels for adaptive teaming without teammate modelling. Red dotted lines denote best-response baselines specifically trained on the given unseen teammate zoo.

ENV	Metrics	Unseen Zoo 1				Unseen Zoo 2				Unseen Zoo 3			
		SP	PBT	H-D w/o g	H-D	SP	PBT	H-D w/o g	H-D	SP	PBT	H-D w/o g	H-D
4p2e3o	COL↓	32.40 ±4.34	25.60 ±15.19	**16.40** ±7.27	22.33 ±6.81	39.20 ±7.35	34.00 ±12.10	28.80 ±4.15	**22.40** ±3.58	31.60 ±4.90	24.00 ±14.38	18.80 ±8.79	**16.80** ±2.28
	AST↓	313.48 ±51.91	303.21 ±86.38	260.42 ±20.68	**259.17** ±34.28	380.28 ±34.65	376.14 ±68.19	329.24 ±23.35	**314.97** ±27.68	380.85 ±34.36	328.96 ±85.25	308.05 ±21.18	**306.35** ±25.86
	REW↑	123.60 ±4.68	132.34 ±22.68	**141.51** ±5.75	138.28 ±7.95	114.00 ±9.07	121.98 ±10.13	130.38 ±4.78	**135.71** ±3.46	126.15 ±4.68	133.11 ±13.39	139.91 ±9.48	**143.55** ±1.29
4p2e1o	COL↓	30.80 ±12.03	30.40 ±5.22	26.00 ±7.35	**19.20** ±7.56	36.80 ±15.59	43.20 ±10.73	36.40 ±8.29	**32.00** ±5.83	34.80 ±14.52	35.60 ±5.40	25.60 ±6.07	**23.60** ±6.54
	AST↓	352.24 ±25.07	317.55 ±30.08	**279.43** ±29.16	298.93 ±29.19	446.67 ±51.69	395.92 ±72.36	383.39 ±43.82	**353.35** ±35.29	360.08 ±22.18	375.41 ±48.52	335.27 ±36.29	**313.60** ±34.18
	REW↑	122.05 ±9.92	126.73 ±6.91	129.97 ±9.73	**138.24** ±6.28	112.66 ±11.94	109.06 ±13.42	114.00 ±10.64	**120.83** ±6.84	118.73 ±13.22	120.00 ±5.91	128.76 ±8.15	**131.50** ±5.76
4p2e5o	COL↓	59.60 ±10.39	53.00 ±2.61	56.00 ±9.49	**49.86** ±11.62	31.20 ±3.42	33.50 ±14.74	30.00 ±12.41	**25.20** ±3.35	44.00 ±5.74	40.50 ±2.45	39.20 ±7.56	**32.00** ±6.16
	AST↓	333.96 ±51.81	**313.53** ±34.83	345.89 ±45.29	331.98 ±80.30	287.18 ±55.20	348.23 ±18.91	321.41 ±83.02	**281.45** ±40.77	294.41 ±34.95	340.94 ±38.79	332.19 ±34.72	**313.79** ±26.78
	REW↑	90.16 ±12.00	93.16 ±7.47	86.30 ±14.37	**99.31** ±17.70	120.48 ±8.33	124.92 ±19.11	122.01 ±12.34	**128.45** ±3.79	107.56 ±5.44	107.08 ±5.92	104.57 ±14.15	**119.93** ±6.11
4p3e5o	COL↓	62.80 ±5.02	64.80 ±4.60	**58.00** ±8.72	67.57 ±15.85	40.40 ±5.83	38.00 ±8.05	40.00 ±2.45	**36.40** ±4.34	55.60 ±8.65	41.60 ±4.77	45.60 ±3.29	**38.40** ±5.18
	AST↓	431.44 ±11.17	509.56 ±67.14	**418.51** ±41.49	459.55 ±91.71	446.07 ±49.11	555.10 ±38.97	482.64 ±82.82	**407.85** ±63.06	425.94 ±38.53	510.96 ±60.10	456.98 ±52.86	**416.91** ±76.21
	REW↑	**141.98** ±15.98	131.29 ±9.66	136.36 ±19.43	116.57 ±43.94	182.04 ±8.97	187.32 ±7.17	185.98 ±4.43	**196.07** ±9.91	146.74 ±17.01	172.06 ±11.93	159.60 ±1.74	**174.59** ±18.68

Table 2: Performance comparison of adaptive teaming without teammate modeling across environments with varying difficulties. H-D denotes HOLA-Drone (V2).

adaptive teaming without explicit teammate modeling, and (2) adaptive teaming incorporating teammate modeling. Each subsection outlines the experimental setups, highlights critical findings, and provides detailed analyses.

Adaptive Teaming without Teammate Modeling. Fig. 2 and Table 2 comprehensively evaluate four baseline methods—SP, PBT, HOLA-Drone (V2) w/o g, and HOLA-Drone (V2)—across four progressively challenging multi-drone pursuit scenarios (4p2e3o, 4p2e1o, 4p2e5o, and 4p3e5o). The HOLA-Drone (V2) w/o g variant serves as an ablation study, removing the core hypergraphic game mechanism to assess its impact. Each scenario is tested against three distinct unseen teammate zoos. Fig. 2 primarily highlights success rates (SUC), while Table 2 further details collision counts (COL), average steps taken (AST), and cumulative rewards (REW). Together, these metrics demonstrate the efficacy of adaptive teaming methods within the AT-Drone benchmark.

Red dotted lines in Fig. 2 denote approximate upper-bound performances established by best-response (BR) policies specifically trained for each unseen teammate zoo, serving as critical reference points for evaluating adaptive teaming effectiveness. As environmental complexity escalates from the simplest scenario (4p2e1o) to the most complex (4p3e5o), success rates across all evaluated methods decrease, and the performance gap compared to BR policies widens. This clearly illustrates the increasing challenge of effective coordination with unfamiliar teammates.

Overall, HOLA-Drone (V2) consistently demonstrates superior performance across most adaptive teaming scenarios, achieving higher success rates, fewer collisions, and reduced average steps compared to other baseline methods. Notably, the performance advantages of HOLA-Drone (V2)

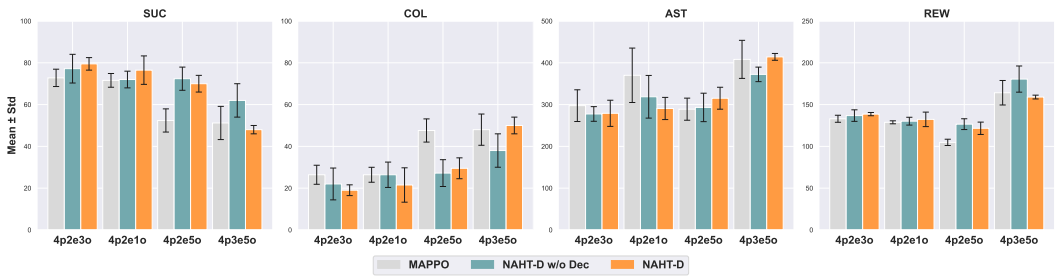

Figure 3: Performance comparison of adaptive teaming with teammate modeling across environments with varying difficulties.

become significantly more pronounced in complex scenarios, underscoring the value of advanced adaptive strategies in dynamic environments involving diverse teammate behaviors.

Interestingly, the simplified ablation model, HOLA-Drone (V2) w/o g—lacking the core hypegraphical-form game module—achieves comparable or even superior performance metrics in simpler scenario 4p2e1o. This observation suggests that while the hierarchical graphical module greatly benefits coordination in challenging environments, it may introduce unnecessary complexity when facing relatively straightforward conditions.

Adaptive teaming with teammate modelling. Fig. 3 illustrates the comparative performance of MAPPO, NAHT-D w/o Dec, and NAHT-D methods across four multi-drone pursuit scenarios: 4p2e3o, 4p2e1o, 4p2e5o, and 4p3e5o, explicitly incorporating teammate modeling into the adaptive teaming process. The NAHT-D w/o Dec variant serves as an ablation study, removing the teammate modeling decoder network and the corresponding loss function to assess their impact.

Across simpler scenarios (4p2e3o and 4p2e1o), NAHT-D achieves marginally higher or comparable success rates (SUC) compared to NAHT-D w/o Dec, with both outperforming MAPPO. However, as scenario complexity increases (4p2e5o and 4p3e5o), NAHT-D w/o Dec demonstrates notably superior performance over NAHT-D, indicating that the additional complexity introduced by the teammate modeling decoder may negatively impact coordination efficiency under highly challenging conditions. Regarding collision counts (COL) and average steps (AST), NAHT-D w/o Dec consistently achieves better or comparable results compared to NAHT-D, further suggesting that simpler teammate modeling strategies can offer superior robustness and efficiency in complex adaptive teaming scenarios. These insights are also supported by cumulative rewards (REW), where NAHT-D w/o Dec generally achieves higher or similar values across all settings.

Case study and demo videos. Appendix E provides a detailed case study demonstrating the real-world deployment of our adaptive teaming approach. In this case study, ATM learners are paired with unseen drone partners sampled from Zoo 3 and tasked with operating in the most challenging environment 4p3e5o. Further demonstration videos can be accessed on our project website at https://sites.google.com/view/at-drone.

5 Conclusion

In this paper, we introduced AT-Drone, a novel multi-robot collaboration benchmark specifically designed to investigate adaptive teaming problems in multi-drone pursuit scenarios. AT-Drone uniquely integrates customizable simulation environments, real-world deployment pipelines leveraging Crazyflie drones and edge computing, a distributed training framework supporting diverse adaptive teaming algorithms, and standardized evaluation protocols. To facilitate comprehensive evaluation, AT-Drone provides four progressively challenging adaptive teaming environments and a collection of three distinct, unseen drone teammate zoos. Experimental results conducted with Crazyflie drones confirm AT-Drone's effectiveness in driving advancements in adaptive teaming research. Furthermore, real-world drone experiments validate the benchmark's practical feasibility and utility, demonstrating its relevance and applicability for realistic robotic operations.

6 Limitations

While AT-Drone successfully bridges simulation and real-world deployment, the current real-world system remains relatively simple, potentially limiting its ability to capture more intricate scenarios encountered in practical operations. Additionally, scalability constraints posed by physical hardware limitations—such as size, payload restrictions inherent to Crazyflie drones, and the current maximum of four pursuers and two evaders within a 3.6m × 5m area—may restrict larger-scale experimentation, making it challenging to study more complex pursuit tasks. Another critical limitation relates to perception; the current setup may not adequately handle complex perception tasks, potentially limiting the drones' adaptability in visually challenging environments. Despite efforts to replicate realistic scenarios, simulation environments may not fully encapsulate all complexities and uncertainties present in real-world conditions. Future research should focus on addressing these limitations by enhancing system complexity, improving perception capabilities, exploring scalable drone hardware alternatives, enhancing simulation fidelity, and developing methods suitable for larger-scale, more complex scenarios.

Acknowledge

We sincerely thank Jianghong Wang for constructive feedback during the preparation of this manuscript.

References

[1] T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics: A survey. *Autonomous robots*, 31:299–316, 2011.

[2] Z. Zhang, D. Zhang, Q. Zhang, W. Pan, and T. Hu. DACOOP-A: Decentralized adaptive cooperative pursuit via attention. *IEEE Robotics and Automation Letters*, PP:1–8, 11 2023. doi:10.1109/LRA.2023.3331886.

[3] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *Ieee Access*, 8:191617–191643, 2020.

[4] K. Shah and M. Schwager. Multi-agent Cooperative Pursuit-Evasion Strategies Under Uncertainty. In N. Correll, M. Schwager, and M. Otte, editors, *Distributed Autonomous Robotic Systems*, pages 451–468, Cham, 2019. Springer International Publishing. ISBN 978-3-030-05816-6. doi:10.1007/978-3-030-05816-6_32.

[5] M. Janosov, C. Virágh, G. Vásárhelyi, and T. Vicsek. Group chasing tactics: how to catch a faster prey? *New Journal of Physics*, 19(5):053003, May 2017. ISSN 1367-2630. doi:10.1088/1367-2630/aa69e7. URL http://arxiv.org/abs/1701.00284. arXiv:1701.00284 [physics].

[6] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Stipanović, and C. J. Tomlin. Cooperative pursuit with Voronoi partitions. *Automatica*, 72:64–72, Oct. 2016. ISSN 0005-1098. doi:10.1016/j.automatica.2016.05.007. URL https://www.sciencedirect.com/science/article/pii/S0005109816301911.

[7] J. Chen, G. Li, C. Yu, X. Yang, B. Xu, H. Yang, and Y. Wang. A dual curriculum learning framework for multi-uav pursuit-evasion in diverse environments, 2024. URL https://arxiv.org/abs/2312.12255.

[8] S. Qi, X. Huang, P. Peng, X. Huang, J. Zhang, and X. Wang. Cascaded Attention: Adaptive and Gated Graph Attention Network for Multiagent Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):3769–3779, Mar. 2024. ISSN

2162-2388. doi:10.1109/TNNLS.2022.3197918. URL https://ieeexplore.ieee.org/abstract/document/9913678. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[9] C. de Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić. Decentralized Multi-Agent Pursuit Using Deep Reinforcement Learning. *IEEE Robotics and Automation Letters*, 6 (3):4552–4559, July 2021. ISSN 2377-3766. doi:10.1109/LRA.2021.3068952. URL https://ieeexplore.ieee.org/abstract/document/9387125. Conference Name: IEEE Robotics and Automation Letters.

[10] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Russell. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4213–4220, July 2019. ISSN 2374-3468. doi:10.1609/aaai.v33i01.33014213. URL https://ojs.aaai.org/index.php/AAAI/article/view/4327. Number: 01.

[11] L. Matignon, G. J. Laurent, and N. Le Fort-Piat. Hysteretic Q-learning : an algorithm for Decentralized Reinforcement Learning in Cooperative Multi-Agent Teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 64–69, Oct. 2007. doi:10.1109/IROS.2007.4399095. URL https://ieeexplore.ieee.org/abstract/document/4399095. ISSN: 2153-0866.

[12] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pages 4399–4410. PMLR, 2020.

[13] P. Stone, G. Kaminka, S. Kraus, and J. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1504–1509, 2010.

[14] Y. Li, S. Zhang, J. Sun, Y. Du, Y. Wen, X. Wang, and W. Pan. Cooperative open-ended learning framework for zero-shot coordination. In *International Conference on Machine Learning*, pages 20470–20484. PMLR, 2023.

[15] X. Wang, S. Zhang, W. Zhang, W. Dong, J. Chen, Y. Wen, and W. Zhang. Zsc-eval: An evaluation toolkit and benchmark for multi-agent zero-shot coordination. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[16] K. Lucas and R. E. Allen. Any-play: An intrinsic augmentation for zero-shot coordination. In P. Faliszewski, V. Mascardi, C. Pelachaud, and M. E. Taylor, editors, *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, pages 853–861. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022. doi:10.5555/3535850.3535946. URL https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p853.pdf.

[17] R. Canaan, X. Gao, J. Togelius, A. Nealen, and S. Menzel. Generating and adapting to diverse ad hoc partners in hanabi. *IEEE Transactions on Games*, 15(2):228–241, 2022.

[18] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

[19] S. Barrett, P. Stone, and S. Kraus. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 567–574, 2011.

[20] G. Papoudakis, F. Christianos, and S. Albrecht. Agent modelling under partial observability for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 19210–19222, 2021.

[21] C. Wang, M. A. Rahman, I. Durugkar, E. Liebman, and P. Stone. N-agent ad hoc teamwork. *Advances in Neural Information Processing Systems*, 37:111832–111862, 2024.

[22] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[23] Y. Zhang, M. Ding, Y. Yuan, J. Zhang, Q. Yang, G. Shi, F. Jiang, and M. Lu. Multi-uav cooperative pursuit of a fast-moving target uav based on the gm-td3 algorithm. *Drones*, 8(10): 557, 2024.

[24] Y. Li, D. Zhang, J. Chen, Y. Wen, Q. Zhang, S. Mou, and W. Pan. Hola-drone: Hypergraphic open-ended learning for zero-shot multi-drone cooperative pursuit. *CoRR*, abs/2409.08767, 2024. doi:10.48550/ARXIV.2409.08767. URL https://doi.org/10.48550/arXiv.2409.08767.

[25] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

[26] A. Lupu, B. Cui, H. Hu, and J. Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning (ICML)*, pages 7204–7213. PMLR, 2021.

[27] R. Zhao, J. Song, Y. Yuan, H. Hu, Y. Gao, Y. Wu, Z. Sun, and W. Yang. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6145–6153, 2023.

[28] R. Charakorn, P. Manoonpong, and N. Dilokthanakul. Generating diverse cooperative agents by learning incompatible policies. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] C. Yu, J. Gao, W. Liu, B. Xu, H. Tang, J. Yang, Y. Wang, and Y. Wu. Learning zero-shot cooperation with humans, assuming humans are biased. *arXiv preprint arXiv:2302.01605*, 2023.

[30] Y. Li, S. Zhang, J. Sun, W. Zhang, Y. Du, Y. Wen, X. Wang, and W. Pan. Tackling cooperative incompatibility for zero-shot human-ai coordination. *Journal of Artificial Intelligence Research*, 80:1139–1185, 2024.

[31] S. Barrett, A. Rosenfeld, S. Kraus, and P. Stone. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence*, 242:132–171, 2017.

[32] S. Chen, E. Andrejczuk, Z. Cao, and J. Zhang. Aateam: Achieving the ad hoc teamwork by employing the attention mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7095–7102, 2020.

[33] M. A. Rahman, N. Hopner, F. Christianos, and S. V. Albrecht. Towards open ad hoc teamwork using graph-based policy learning. In *International conference on machine learning*, pages 8776–8786. PMLR, 2021.

[34] J. Wang, Y. Li, Y. Zhang, W. Pan, and S. Kaski. Open ad hoc teamwork with cooperative game theory. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[35] S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[36] T. Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[38] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

[39] M. Janosov, C. Virágh, G. Vásárhelyi, and T. Vicsek. Group chasing tactics: How to catch a faster prey. *New Journal of Physics*, 19, 05 2017. doi:10.1088/1367-2630/aa69e7.

[40] D. Balduzzi, M. Garnelo, Y. Bachrach, W. Czarnecki, J. Perolat, M. Jaderberg, and T. Graepel. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning*, pages 434–443. PMLR, 2019.

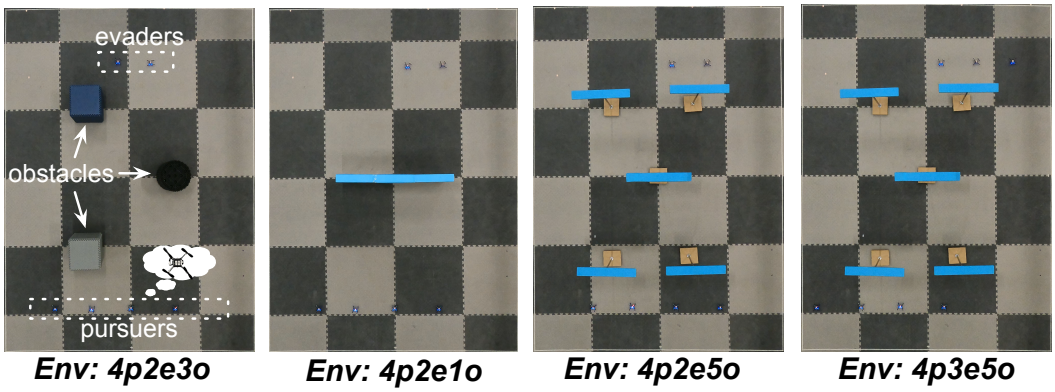

Figure 4: Illustration of four multi-drone pursuit environments in real world. The environments vary in the number of pursuers (p), evaders (e), and obstacles (o), denoted as `4p2e3o`, `4p2e1o`, `4p2e5o`, and `4p3e5o`. Each setup introduces different levels of complexity, testing the adaptability and coordination capabilities of the agents.

A Environment Configurator

The AT-MDP framework environment configurator allows users to define and modify multi-drone pursuit scenarios through a structured JSON file. Fig. 5 provides an example configuration file that specifies key parameters across three categories: *players*, *site*, and *task*.

Players Configuration: This section defines the number and roles of agents in the environment, including the number of pursuers (`num_p`), evaders (`num_e`), controlled agents (`num_ctrl`), and unseen teammates (`num_unctrl`). Additional parameters such as random respawn behavior, reception range, and velocity settings further customize agent interactions. The `unseen_drones` field allows users to specify different unseen teammate models from the unseen drone zoo.

Site Configuration: This section defines the physical properties of the environment, including its boundary dimensions (`width`, `height`) and obstacle placements. Obstacles can be configured individually to introduce varying levels of complexity.

Task Configuration: This section sets the pursuit task parameters, including the capture range (`capture_range`), safety radius (`safe_radius`), task duration (`task_horizon`), and simulation frame rate (`fps`). The `task_name` field provides a label for different predefined environment scenarios.

This modular configuration enables flexible environment customization, facilitating experiments across diverse multi-drone pursuit scenarios.

B Unseen Drone Zoo

Rule-Based Method: Greedy Drone. The Greedy Drone pursues the closest target by continuously aligning its movement with the target's position. Its state information includes its own position, orientation, distances and angles to teammates and evaders, and proximity to obstacles or walls. When obstacles or other agents enter its evasion range, the Greedy Drone dynamically adjusts its direction to avoid collisions, prioritising immediate objectives over team coordination.

Traditional Method: VICSEK Drone. Based on the commonly used VICSEK algorithm [39, 2, 24], the VICSEK Drone adopts a bio-inspired approach to mimic swarm-like behaviours. It computes and updates a velocity vector directed towards the evader, optimising the tracking path based on the agent's current environmental state. To avoid nearby obstacles or agents, the VICSEK Drone applies repulsive forces with varying magnitudes. While the calculated velocity vector includes

```
{
    "players": {
        "num_p": 4,
        "num_e": 2,
        "num_ctrl": 2,
        "num_unctrl": 2,
        "random_respawn": True,
        "respawn_region":  {***},
        "reception_range": 2,
        "velocity_p": 0.3,
        "velocity_e": 0.6,
        "unseen_drones": [***]
    },
    "site":{
        "boundary": {
            "width" : 3.6,
            "height" : 5,
        },
        "obstacles": {
            "obstacle1":{***}
        },
    },
    "task":{
        "task_name": 4p2e1o,
        "capture_range": 0.2,
        "safe_radius": 0.1,
        "task_horizon": 100,
        "fps": 10,
    }
}
```


Figure 5: An example of environment configuration file.

both magnitude and orientation, only the orientation is implemented in our experiments, making it a scalable and practical teammate model for multi-drone coordination.

Learning-Based Method: Self-Play Drones. For the learning-based approach, we employ an IPPO-based self-play algorithm, generating diverse drone behaviours by training agents with different random seeds. This approach simulates a wide range of adaptive strategies, introducing stochasticity and complexity to the evaluation process.

C HOLA-Drone (V2) Algorithm

In this section, we define a population of drone strategies, denoted as $\Pi = \{\pi_1, \pi_2, \cdots, \pi_n\}$. For the task involving C teammates, the interactions within the population Π are modeled as a hypergraph $\mathcal{G}$. Formally, the hypergraph is represented by the tuple $(\Pi, \mathcal{E}, \mathbf{w})$, where the node set Π represents the strategies, $\mathcal{E}$ is the hyperedge set capturing interaction relationships among teammates, and $\mathbf{w}$ is the weight set representing the corresponding average outcomes. The left subfigure of Fig. 6 illustrates an example of a hypergraph representation with five nodes and a fixed hyperedge length of 4.

Building on the concept of preference hypergraphs [24], we use the preference hypergraph to represent the population and assess the coordination ability of each node. The **preference hypergraph** $\mathcal{PG}$ is derived from the hypergraph $\mathcal{G}$, where each node has a direct outgoing hyperedge pointing to the teammates with whom it achieves the highest weight in $\mathcal{G}$. Formally, $\mathcal{PG}$ is defined by the tuple $(\Pi, \mathcal{E}_\mathcal{P})$, where the node set Π represents the strategies, and $\mathcal{E}_\mathcal{P}$ denotes the set of outgoing hyperedges. As shown in the right subfigure of Fig. 6, the dotted line highlights the outgoing edge. For instance, node 2 has a single outgoing edge $(2, 3, 5, 4)$ because it achieves the highest outcome, i.e., a weight of 45, with those teammates in $\mathcal{G}$, as depicted in the left subfigure.

Intuitively, a node in $\mathcal{PG}$ with higher cooperative ability will have more incoming hyperedges, as other agents prefer collaborating with it to achieve the highest outcomes. Therefore, we extend the concept of **preference centrality** [14] to quantify the cooperative ability of each node. Specifically, for any node $i \in \Pi$, the preference centrality is defined as

$$\eta_{\Pi}(i) = \frac{d_{\mathcal{PG}}(i)}{d_{\mathcal{G}}(i)}, \tag{1}$$

where $d_{\mathcal{PG}}(i)$ denotes the incoming degree of node i in $\mathcal{PG}$, and $d_{\mathcal{G}}(i)$ represents the degree of node i in $\mathcal{G}$.

Max-Min Preference Oracle. Building on the basic definition of the preference hypergraph representation, we introduce the concept of Preference Optimality to describe the goal of our training process.

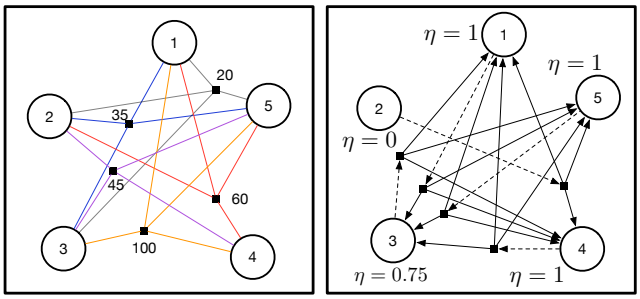

Figure 6: An example of a hypergraph representation (left) and its corresponding preference hypergraph (right) with five strategies in the population.

Definition C.1 (Preference Optimal)**.** A set of learners $\mathcal{N}^{\star}$ of size N is said to be **Preference Optimal (PO)** in a hypergraph $\mathcal{G} = (\Pi, \mathcal{E}, \mathbf{w})$ if, for any set $\hat{\mathcal{N}} \subseteq \Pi$ of size N, the following condition holds:

$$\sum_{s \in \mathcal{N}^{\star}} \eta_{\Pi}(s) \geq \sum_{s \in \hat{\mathcal{N}}} \eta_{\Pi}(s), \tag{2}$$

where $\eta_{\Pi}(s)$ denotes the preference centrality of learner s in the hypergraph $\mathcal{G}$.

While achieving a preference-optimal oracle is desirable, it becomes impractical or prohibitively expensive in large, diverse populations. Therefore, we propose the **max-min preference oracle**, abbreviated as *oracle* in the rest of this paper, to ensure robust adaptability and maximize cooperative performance under the worst-case teammate scenarios.

To formalize the objective, we split the strategy population Π into a learner set $\mathcal{N}$ and a non-learner set $\Pi_{-\mathcal{N}}$, where $\Pi_{-\mathcal{N}} \cap \mathcal{N} = \emptyset$ and $\Pi_{-\mathcal{N}} \cup \mathcal{N} = \Pi$. The objective function ϕ is defined as:

$$\phi : \underbrace{\mathcal{N} \times \cdots \times \mathcal{N}}_{N \text{ learners}} \times \underbrace{\Pi_{-\mathcal{N}} \times \cdots \times \Pi_{-\mathcal{N}}}_{M \text{ teammates}} \to \mathbb{R}. \tag{3}$$

The max-min preference oracle updates the learner set by solving:

$$\mathcal{N}' = oracle(\mathcal{N}, \phi_{\mathcal{M}}(\cdot)) := \arg\max_{\mathcal{N}} \min_{\mathcal{M} \subseteq \Pi_{-\mathcal{N}}} \phi_{\mathcal{M}}(\mathcal{N}), \tag{4}$$

where the objective $\phi_{\mathcal{M}}(\cdot)$ is derived using the extended curry operator [40], originally designed for two-player games, and is expressed as:

$$\begin{aligned} &\left[\underbrace{\mathcal{N} \times \cdots \times \mathcal{N}}_{N \text{ learners}} \times \underbrace{\Pi_{-\mathcal{N}} \times \cdots \times \Pi_{-\mathcal{N}}}_{M \text{ teammates}} \to \mathbb{R} \right] \\ \to &\left[\underbrace{\Pi_{-\mathcal{N}} \times \cdots \times \Pi_{-\mathcal{N}}}_{M \text{ teammates}} \to \left[\underbrace{\mathcal{N} \times \cdots \times \mathcal{N}}_{N \text{ learners}} \to \mathbb{R} \right] \right]. \end{aligned} \tag{5}$$

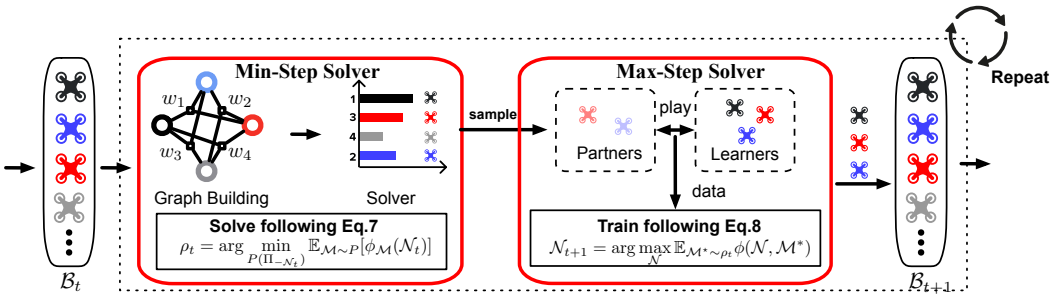

Figure 7: Overview of our proposed HOLA-Drone (V2) algorithm.

Intuitively, *oracle* alternates between two key steps: the minimization step and the maximization step. In the minimization step, the objective is to identify the subset of teammates $\mathcal{M}^* \subset \Pi_{-\mathcal{N}}$ that minimizes the performance outcome of the current learner set $\mathcal{N}$, i.e. the worst partners. This is formulated as:

$$\mathcal{M}^* = \arg \min_{\mathcal{M}\subset\Pi_{-\mathcal{N}}} \phi_{\mathcal{M}}(\mathcal{N}).$$

In the maximization step, the learner set $\mathcal{N}$ is updated to maximize its performance outcome against the identified subset $\mathcal{M}^*$. This is defined as:

$$\mathcal{N}^* = \arg\max_{\mathcal{N}} \phi(\mathcal{N}, \mathcal{M}^*).$$

To achieve robust adaptability and dynamic coordination in multi-agent systems, we integrate the max-min preference oracle into an open-ended learning framework, referred to as the HOLA-Drone (V2) algorithm. The HOLA-Drone (V2) algorithm dynamically adjusts the training objective as the population evolves, enabling continuous improvement and effective coordination with unseen partners. Unlike conventional fixed-objective training, the HOLA-Drone (V2) approach iteratively expands the strategy population Π and refines the learner set $\mathcal{N}$. At each generation t, the framework recalibrates the training objective ϕ based on new extended population Π_t to account for the evolving interactions among agents within the population.

As shown in Fig. 6, HOLA-Drone (V2) algorithm consists of two key modules: the min-step solver and the max-step trainer. At each generation t, the updated learner set $\mathcal{N}_t$ from the previous generation $t-1$ is incorporated into the population Π_{t-1}, resulting in an expanded population Π_t.

Min-step Solver. The role of the min-step solver is to first construct the preference hypergraph representation of the interactions within the updated population Π_t. Here, we only need to build a subgraph of the entire preference hypergraph in Π_t, denoted as $\mathcal{PG}'_t$. To obtain $\mathcal{PG}'_t$, we focus on constructing the hyperedges in the hypergraph $\mathcal{G}'_t$ that connect to the learner set $\mathcal{N}_t$. For instance, if Π_t consists of a learner set $\mathcal{N}_t$ of size N and a non-learner set $\Pi_{-\mathcal{N}_t}$, any hyperedge e in $\mathcal{G}'_t$ connects N nodes from $\mathcal{N}_t$ and all possible M nodes from $\Pi_{-\mathcal{N}_t}$. The preference hypergraph $\mathcal{PG}'_t$ is then derived from $\mathcal{G}'_t$ by retaining only the outgoing hyperedge with the highest weight for each node.

The min-step solver uses the reciprocal of the preference centrality in $\mathcal{PG}'_t$ to evaluate the worst-case partners. To enhance robustness, the solver does not deterministically select the worst-case partners as $\mathcal{M}^* = \arg\min_{\mathcal{M}\subset\Pi_{-\mathcal{N}}} \phi_{\mathcal{M}}(\mathcal{N})$. Instead, it outputs a mixed strategy ρ_t, defined as:

$$\rho_t = \arg \min_{P(\Pi_{-\mathcal{N}_t})} \mathbb{E}_{\mathcal{M}\sim P}[\phi_{\mathcal{M}}(\mathcal{N}_t)]. \tag{6}$$

In practice, the mixed strategy ρ_t is obtained by normalizing the reciprocal of the preference centrality, assigning higher probabilities to worse partners.

Max-Step Solver. Given the mixed strategies ρ_t, the max-step solver iteratively samples the worst-case partners, referred to as the profile, $\mathcal{M} \sim \rho_t$, from the non-learner set $\Pi_{-\mathcal{N}_t}$. It simulates interactions between the sampled profile and the learners to generate training data, with the objective of maximizing the reward $\phi(\mathcal{N}_t, \mathcal{M}) = \mathbb{E}_{\mathcal{N}_t,\mathcal{M}}[R(\tau)]$, as shown in Eq. 3.1. The max-step oracle

Table 3: Implementation hyperparameters of NAHT -D algorithm.

Parameters	Values	Parameters	Values
Batch size	1024	Minibatch size	256
Lambda (λ)	0.99	Generalized advantage estimation lambda (λ_{gae})	0.95
Learning rate	3e-4	Value loss coefficient(c_1)	1
Entropy coefficient(ϵ_{clip})	0.01	PPO epoch	20
Total environment step	1e6	History length	1
Embedding size	16	Hidden size	128

could be rewritten as

$$\mathcal{N}_{t+1} = \arg\max_{\mathcal{N}} \mathbb{E}_{\mathcal{M}^{\star}\sim\rho_t}\phi(\mathcal{N}, \mathcal{M}^{*}). \tag{7}$$

The strategy network for adaptive teaming, supports both AHT and ZSC paradigms. In the AHT paradigm, the network uses a teammate modeling network (f) to infer teammate types from the observation history (τ_t^i). These predicted vector, combined with the agent's observation history (τ_t^i), are input into a PPO-based policy network. This policy network includes an Actor Network (π_θ) for generating the agent's action (a_t^i) and a Critic Network (V_π) for evaluating the policy.

In contrast, the ZSC paradigm simplifies the process by directly feeding the agent's observation history (τ_t^i) into the actor and critic networks, bypassing explicit teammate modeling. This approach enables the agent to coordinate with unseen teammates without prior knowledge or additional inference mechanisms.

The max-step solver ultimately generates an approximate best response $\mathcal{N}_{t+1}$ to the worst-case partners, enhancing the agent's adaptive coordination capabilities.

D NAHT-D algorithm

NAHT-D extends the MAPPO algorithm [38] by incorporating an additional teammate modeling network f. This network generates team encoding vectors to represent the behavioral characteristics of unseen teammates, improving coordination in multi-drone pursuit. The modeling network f processes three types of inputs: (1) observed evader states history, (2) self-observed states history, and (3) relative positions history between agents. These inputs are transformed using independent fully connected layers, aggregated through weighted averaging, and combined into a unified team representation. The final embedding is a fixed-dimensional vector, integrated into the actor network of MAPPO to enhance decision-making in adaptive teaming scenarios. The details of hyperparameters are listed in Table 3.

E Case Study

To further illustrate the effectiveness of our adaptive teaming approach, we present a case study in the `4p3e5o` environment, categorized as superhard due to its high complexity, featuring four pursuers, three evaders, and five obstacles. The unseen teammates in this scenario are sampled from Unseen Zoo 3, which consists entirely of PPO-based self-play policies trained at two different skill levels, introducing high adaptability and unpredictability.

This case study demonstrates how the NAHT-D learners effectively coordinate with their unseen drone partners to execute a multi-stage capture strategy. Fig. 8 illustrates key frames from the scenario. In Frame 1, four pursuers initiate a collaborative approach, positioning themselves strategically to encircle all three evaders while maintaining an adaptive formation. In Frame 2, two pursuers successfully capture one of the evaders while the other two tighten their formation, preventing the remaining evaders from escaping. In Frames 3 & 4, the remaining two evaders are captured one by one as the pursuers continue refining their positioning and coordination, effectively closing all escape routes.

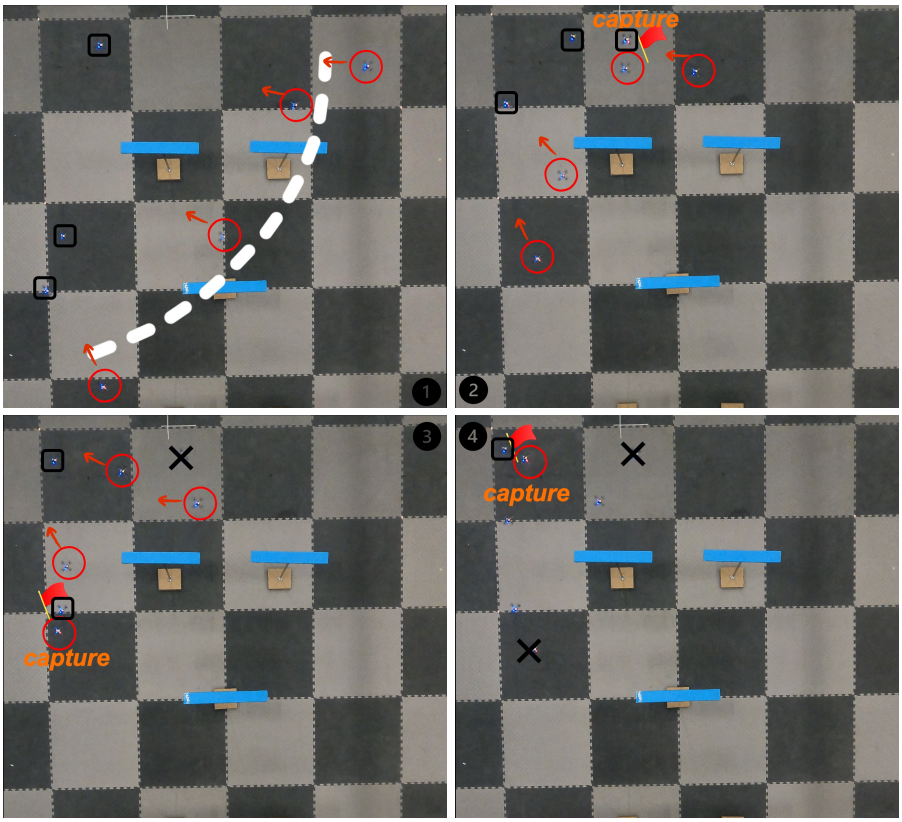

Figure 8: **Case Study:** This example demonstrates the capture strategy executed by NAHT-D learners and unseen drone partners from unseen zoo 3 in the superhard environment 4p3e5o. The red circles denote pursuers, and the black squares represent evaders. In this scenario, four pursuers collaboratively surround all three evader (1), two pursuers capture one of evaders while other two pursuers continuously tighten their formation (2), and rest of two evaders are then successfully captured one by one (3 & 4)