Journal of Industrial and Management Optimization

https://www.aimspress.com/journal/jimo

JIMO, 22(12): 6213–6247.
DOI:10.3934/jimo.2026209
Received: 16 September 2026
Revised: 22 October 2026
Accepted: 03 November 2026
Published: 06 November 2026

Research article

HOCCA: Hierarchical option-based communication-constrained attention for cooperative multi-UAV target search

Jiancheng Liu[1], **Wanlong Qi**[4], **Yihua Wang**[1], **Bo Li**[1,2], **Kexin Wang**[5,*] **and Dingrui Xue**[3,*]

[1] Northwest Institute of Mechanical and Electrical Engineering, Xianyang 712099, Shaanxi, China

[2] School of Computer Science and Technology, Tsinghua University, Beijing 100084, China

[3] School of Information Engineering, Chang'an University, Xi'an 710064, Shaanxi, China

[4] School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China

[5] School of Computer Science and Technology, xidian University, Xi'an 710021, Shaanxi, China

* **Correspondence:** Email: kexinwang0618@stu.xidian.edu.cn; drxue@chd.edu.cn.

Abstract: Efficient cooperative target search with multiple unmanned aerial vehicles (UAVs) is fundamentally constrained by partial observability, sensing uncertainty, and limited interagent communication. We formulate this task as a communication-constrained decentralized partially observable Markov decision process with explicit detection and false-alarm models and target-motion prediction. To address these challenges, we propose hierarchical option-based communication-constrained attention (HOCCA), a multiagent reinforcement learning framework that integrates temporally extended options, message-restricted graph attention, and value-of-information-driven communication under centralized training and decentralized execution. Each UAV maintains a provenance-aware occupancy belief using local observations and received likelihood innovations, while unique identifiers prevent duplicate evidence fusion. A receiver-aware scheduler selects informative evidence subject to a hard communication budget, and the hierarchical controller jointly learns option selection, termination, and intraoption maneuvering. In the reference environment, HOCCA requires 25.1% and 23.8% fewer steps than the strongest hierarchical baseline to achieve 80% coverage and detection, respectively, while recording the highest coverage and recall at 260 steps. Evaluations across varying target densities, map scales, and fleet sizes, together with a representative case study, demonstrate efficient coordination across the tested task configurations.

Keywords: Multi-UAV cooperative search; hierarchical reinforcement learning; option-critic; multiagent reinforcement learning; belief space; selective communication; uncertain sensing
Mathematics Subject Classification: 90C40, 68T07, 93E35, 90B40

1. Introduction

Cooperative searching with multiple unmanned aerial vehicles (UAVs) enables time-critical sensing missions to be distributed across multiple mobile platforms, thereby improving spatial coverage and operational efficiency. Representative applications include post-disaster victim localization, wildfire monitoring, maritime patrol, and battlefield surveillance [1, 2]. Effective coordination, however, remains challenging because each UAV observes only a limited region through an imperfect sensor, inter-UAV information exchange is constrained by the wireless bandwidth, and the fleet must balance global coverage with targeted investigation under finite time and energy budgets.

Classical search theory allocates sensing effort according to prescribed detection models [3, 4]. Although these formulations provide useful analytical references, they do not directly address online coordination among agents with decentralized and evolving information. Heuristic and swarm-based planners offer computationally efficient decentralized control, but they may converge to locally inefficient search patterns and generally lack a shared belief space objective [5]. Game-theoretic and auction-based methods provide coordination guarantees under specific payoff and information structures [1], whereas offline metaheuristic methods typically require reoptimization when the mission configuration changes.

Multiagent reinforcement learning (MARL) provides a data-driven alternative in which coordinated policies are learned from repeated interaction with a simulated environment. In particular, centralized training with decentralized execution (CTDE) allows joint information to guide policy optimization while preserving decentralized operation. Representative CTDE methods, including multiagent deep deterministic policy gradient (MADDPG) [6], multiagent proximal policy optimization (MAPPO) [7], and QMIX (monotonic mixing of individual action values) [8], have been applied to cooperative multi-UAV searching [1, 9, 10]. Recent studies have further incorporated belief maps [11], graph attention [12, 13], semantic memory [14], language-model-assisted coordination [15], and domain randomization [16]. Despite this progress, three interdependent limitations remain.

First, primitive action policies operate at a single decision timescale, requiring strategic task allocation and reactive maneuvering to be represented within the same action process; the options framework and the option-critic architecture instead introduce temporally extended policies with explicit termination mechanisms [17, 18]. Although dual-timescale hierarchical MADDPG (DTH-MADDPG) separates region-level coordination from low-level control through a dual-timescale hierarchy, it does not formulate high-level decisions as learned options with state-dependent termination [2]. Second, target-search sensors may produce both missed and false detections, which must be incorporated into probabilistic belief updates because they directly affect target estimates and subsequent search actions [19]. Third, representative MARL communication methods provide continuous message aggregation, learned addressing, or bandwidth-aware sender scheduling [20–22]; however, their published formulations do not rank provenance-level evidence according to its marginal information value for an acknowledged receiver belief. These limitations are coupled in decentralized partially observable control. Each agent's action depends on its local information state, while that state evolves through sensing and communicated observations [20, 19].

To address these challenges, we propose HOCCA (hierarchical option-based communication-constrained attention), a hierarchical MARL framework for cooperative multi-UAV searching under sensing and communication uncertainty. We formulate the task as a communication-constrained decentralized partially observable Markov decision process (Dec-POMDP)

with an explicit multitarget Bayesian information state. An exact joint belief is retained for probabilistic analysis, while a factorized occupancy approximation enables tractable policy execution. Each UAV selects temporally extended options from its local history and received messages, executes primitive actions through an intraoption policy, and reselects an option according to a learned termination function. The central distinction from prior CTDE and communication architectures is that coordination is driven by provenance-level, receiver-specific information value under a hard item budget, rather than by undifferentiated message passing or sender scheduling alone. Centralized critics are used exclusively during training.

The main contributions are summarized as follows.

1. We formulate probabilistic sensing with missed detections and false alarms, use provenance-tagged innovations to prevent duplicate fusion, and distribute information gain among UAVs while preserving the aggregate team reward.
2. We develop decentralized option actors, intraoption policies, and learned termination functions that separate long-horizon coordination from primitive UAV maneuvering.
3. We introduce a receiver-aware value-of-information (VoI) scheduler with exact joint receiver-and-item selection under the adopted equal-size modular objective and packet/payload constraints.
4. We provide assumption-explicit analytical results and evaluate HOCCA through baseline comparisons, ablations, generalization studies, scalability tests, and robustness analyses. We also report the architectural and scheduling complexity that governs execution and centralized-training scaling.

2. Related work

2.1. Cooperative multi-UAV search

Classical search theory provides a mathematical foundation for allocating search effort under a prescribed detection model [3, 4]. In multi-UAV systems, this problem is commonly reformulated as coverage planning, spatial partitioning, or distributed task allocation. Artificial potential fields, flocking strategies, and ant-colony-based methods enable computationally efficient decentralized motion, but may exhibit oscillatory behavior or converge to locally inefficient search patterns in complex environments [5, 9]. Probabilistic search and tracking methods instead maintain their target beliefs and explicitly account for false positive and false negative observations [19]. Game-theoretic and auction-based approaches further coordinate agents' decisions through utility optimization, although their performance depends on the adopted payoff structure and information assumptions [1]. Deep reinforcement learning has also been applied to electroencephalography (EEG)-based attention monitoring [21, 22] and threat/anomaly detection in wearable devices [23], illustrating learning-based inference from uncertain sensor data. Recent studies have addressed vision-driven UAV swarm planning [24], bandwidth-constrained coordination of unmanned surface vehicles (USVs) [25], and energy-aware UAV relay optimization [26], connecting cooperative control to perception and communication resources. These methods provide effective spatial coordination mechanisms, but the joint treatment of adaptive control, sensing uncertainty, and bandwidth-constrained information exchange remains limited.

2.2. MARL for multi-UAV search

MARL enables cooperative policies to be learned directly from interactions with uncertain and dynamic environments. The CTDE paradigm allows centralized value functions to exploit joint information during training, while each actor operates from locally available observations during execution [6–8]. Existing applications include multi-UAV trajectory planning under computation and communication constraints [1, 27], dynamic target searching using MAPPO [15], and graph-based coordination for heterogeneous UAV fleets [13, 12]. To address partial observability, belief-driven policies represent uncertainty over the targets' locations rather than relying solely on instantaneous detections [11], whereas semantic episodic memory supports information fusion under limited communication ranges [14]. Language-model-assisted coordination and domain randomization have also been investigated to improve planning and transfer across unseen environments [15, 16]. Nevertheless, most existing approaches operate at the primitive action timescale and treat belief estimation, communication, and temporal abstraction as largely separate components.

2.3. Hierarchical reinforcement learning and temporal abstraction

Hierarchical reinforcement learning addresses long-horizon decision-making by organizing control across multiple temporal scales. The options framework defines a temporally extended action through an initiation set, an intraoption policy, and a termination condition [17]. Representative deep hierarchical methods include hierarchical deep Q-network (DQN) methods [28], FeUdal networks [29], and option-critic, which learns intraoption policies and termination functions jointly [18]. Temporally extended macro-actions have also been incorporated into macro-action Dec-POMDP (MacDec-POMDP) formulations for multirobot coordination under uncertainty [30]. In multi-UAV searches, hierarchical methods commonly separate region-level assignment from local trajectory generation [2, 5]. In particular, DTH-MADDPG uses a slow high-level coordinator and fast low-level heading controllers [2]. By comparison, HOCCA learns persistent option policies and state-dependent termination functions within an explicit belief and communication model. This distinction motivates the inclusion of DTH-MADDPG as the principal hierarchical baseline.

2.4. Positioning and novelty

Learned communication allows agents to construct task-dependent messages rather than relying on fixed protocols. CommNet performs differentiable continuous message aggregation [31], while targeted multiagent communication (TarMAC) learns both message content and intended recipients through an attention-based mechanism [32]. SchedNet also introduces learned sender scheduling for shared communication channels with limited bandwidth [33]. Dynamic size message scheduling (DSMS) handles messages of dynamic size under a bandwidth constraint [34], HiComm retrieves structured hierarchical observations through receiver queries [35], and information bottleneck methods compress messages for communication-efficient coordination [36]. In multi-UAV systems, semantic episodic memory has been used to exchange belief summaries [14], and spatiotemporal graph models have been developed to coordinate agents under communication range constraints [13]. These approaches demonstrate the value of selective information exchange, but their published formulations do not jointly evaluate individual items of evidence against an acknowledged receiver belief, retain measurements' provenance, and enforce exactly-once fusion. HOCCA addresses these requirements by assigning receiver-specific information

scores to provenance-tagged innovations and selecting messages under a hard item budget. Its message-restricted graph attention module processes only locally available features and successfully delivered messages, thereby preserving the decentralized information structure during execution.

HOCCA formulates a cooperative search as belief-driven information allocation under communication constraints by integrating temporal abstraction, provenance-aware Bayesian filtering, and receiver-specific evidence valuation into a unified decentralized control framework. Different from DTH-MADDPG, which separates region-level assignment from low-level maneuver control, HOCCA learns persistent options with state-dependent termination over probabilistic occupancy beliefs. Compared with communication architectures such as TarMAC, SchedNet, DSMS, and HiComm, which primarily optimize message generation, recipient selection, or transmission scheduling, HOCCA evaluates each measurement innovation according to its marginal information value for the intended receiver's acknowledged belief under a hard item budget. Moreover, unlike belief-based MARL approaches that mainly use belief representations as policy inputs, HOCCA exploits belief states to quantify transmission utility and guarantee exactly-once evidence fusion. Therefore, bandwidth limitation is transformed from a message count constraint into an information efficiency objective.

Table 1 contrasts the hierarchy and communication decisions of related methods. HOCCA couples learned temporal persistence with receiver-specific evidence valuation and provenance-aware fusion.

Table 1. Methodological comparison of temporal abstraction and selective communication.

Method	Hierarchy/control	Communication and information allocation
Option-critic [18]	Learned options and termination	Single-agent temporal abstraction
DTH-MADDPG [2]	Region assignment and maneuver control	Spatial coordination at two timescales
TarMAC [32]	Primitive step policies	Learned message content and recipient attention
SchedNet [33]	Primitive step policies	Learned sender scheduling under bandwidth limits
DSMS [34]	Primitive step policies	Dynamic message-size allocation
HiComm [35]	Hierarchical observation retrieval	Receiver queries for structured information
HOCCA	Learned options and termination	Receiver-conditioned innovation scores; packet/payload limits; exactly-once evidence fusion

3. Problem formulation

Let $\mathcal{D} = \{1, \dots, G\}^2$ be a $G \times G$ search grid, $\mathcal{U} = \{1, \dots, N\}$ be the UAV set, and $\mathcal{K} = \{1, \dots, K\}$ be the target set. The location of target k at step t is $Y_{k,t} \in \mathcal{D}$, and the occupancy of cell $c \in \mathcal{D}$ is

$$X_{c,t} = \mathbf{1}\{\exists k \in \mathcal{K} : Y_{k,t} = c\}. \tag{3.1}$$

Here, $X_{c,t} \in \{0, 1\}$ and $\mathbf{1}\{\cdot\}$ is the indicator function.

Target motion is represented by the row-stochastic kernel

$$M_{k,t}(c, c') = \Pr(Y_{k,t+1} = c' \mid Y_{k,t} = c), \qquad \sum_{c' \in \mathcal{D}} M_{k,t}(c, c') = 1. \tag{3.2}$$

For a mobile target at a scheduled motion step, the kernel is

$$M_{k,t}(c, c') = \begin{cases} 1 - p_m, & c' = c, \\ p_m/|\mathcal{N}_8(c)|, & c' \in \mathcal{N}_8(c), \\ 0, & \text{otherwise}, \end{cases} \tag{3.3}$$

where $p_m \in [0, 1]$ is the hop probability and $\mathcal{N}_8(c) \subset \mathcal{D}$ is the set of valid eight-connected neighbors of c. At all other steps, and for static targets, $M_{k,t}$ is the identity kernel. Conditional on the current target configuration, target transitions are independent, so

$$\Pr(\mathbf{Y}_{t+1} = \mathbf{y}' \mid \mathbf{Y}_t = \mathbf{y}) = \prod_{k \in \mathcal{K}} M_{k,t}(y_k, y'_k), \tag{3.4}$$

where $\mathbf{Y}_t = (Y_{1,t}, \ldots, Y_{K,t})$, while $\mathbf{y} = (y_1, \ldots, y_K)$ and $\mathbf{y}' = (y'_1, \ldots, y'_K)$ denote the current and next realizations. The prescribed kernel encodes intermittent local motion through T_m and p_m and supplies the prediction step of the occupancy filter.

The environment state is

$$s_t^{\text{env}} = (\mathbf{Y}_t, \mathbf{P}_t, \boldsymbol{\psi}_t, \mathbf{e}_t, \mathbf{v}_t), \tag{3.5}$$

where $\mathbf{P}_t = (\mathbf{p}_{1,t}, \ldots, \mathbf{p}_{N,t})$ contains UAV positions, $\boldsymbol{\psi}_t = (\psi_{1,t}, \ldots, \psi_{N,t})$ contains headings, $\mathbf{e}_t = (e_{1,t}, \ldots, e_{N,t})$ contains the remaining energy, and $\mathbf{v}_t \in \{0, 1\}^{G^2}$ is the fleet-level visited cell mask. UAV i selects the primitive turn command

$$a_{i,t} \in \mathcal{A} = \{-30^\circ, 0^\circ, +30^\circ\}, \tag{3.6}$$

and moves one cell along the resulting heading, subject to the map boundary and collision rules of the environment.

The sensing footprint of UAV i is

$$\mathcal{F}_i(t) = \{c \in \mathcal{D} : \|\mathbf{p}_{i,t} - c\|_2 \le R_s\}, \tag{3.7}$$

where $R_s > 0$ is the sensing radius. For every $c \in \mathcal{F}_i(t)$, the binary measurement $Z_{i,c,t} \in \{0, 1\}$ satisfies

$$\begin{aligned} \Pr(Z_{i,c,t} = 1 \mid X_{c,t} = 1) &= p_D, \\ \Pr(Z_{i,c,t} = 1 \mid X_{c,t} = 0) &= p_{FA}, \end{aligned} \tag{3.8}$$

where $0 < p_{FA} < p_D < 1$ are the false alarm and detection probabilities. The corresponding Bernoulli likelihood is

$$L(z \mid x) = \left[p_D^z (1 - p_D)^{1-z}\right]^x \left[p_{FA}^z (1 - p_{FA})^{1-z}\right]^{1-x}, \tag{3.9}$$

where $x, z \in \{0, 1\}$. Distinct sensor measurements are assumed to be conditionally independent, given the target configuration. The provenance identifiers ensure that copies of one measurement contribute a single likelihood factor.

The exact belief makes the partial observability explicit and provides the reference distribution from which the operational occupancy filter is derived. Let $\mathcal{I}_t^-$ contain the centralized poses, actions, and

unique evidence available immediately before the step-t measurement update, and let $\mathcal{I}_t^+$ additionally contain the unique measurements in $\mathcal{O}_t$. The predicted and posterior joint target beliefs are

$$\begin{aligned} q_t^-(\mathbf{y}) &= \sum_{\bar{\mathbf{y}} \in \mathcal{D}^K} q_{t-1}^+(\bar{\mathbf{y}}) \prod_{k \in \mathcal{K}} M_{k,t-1}(\bar{y}_k, y_k), \\ \Lambda_t(\mathbf{y}) &= \prod_{(i,c) \in \mathcal{O}_t} L(Z_{i,c,t} \mid x_c(\mathbf{y})), \\ q_t^+(\mathbf{y}) &= \frac{q_t^-(\mathbf{y}) \Lambda_t(\mathbf{y})}{\sum_{\mathbf{y}' \in \mathcal{D}^K} q_t^-(\mathbf{y}') \Lambda_t(\mathbf{y}')}. \end{aligned} \tag{3.10}$$

Here, $q_t^-(\mathbf{y}) = \Pr(\mathbf{Y}_t = \mathbf{y} \mid \mathcal{I}_t^-)$ and $q_t^+(\mathbf{y}) = \Pr(\mathbf{Y}_t = \mathbf{y} \mid \mathcal{I}_t^+)$, $\mathcal{O}_t \subseteq \mathcal{U} \times \mathcal{D}$ is the set of observed UAV–cell pairs, and $x_c(\mathbf{y}) = \mathbf{1}\{\exists k : y_k = c\}$.

Because $|\mathcal{D}|^K$ grows exponentially in K, the controller retains cell and target marginals rather than q_t^+. These marginals are

$$\begin{aligned} b_{c,t}^+ &= \sum_{\mathbf{y} \in \mathcal{D}^K} x_c(\mathbf{y}) q_t^+(\mathbf{y}), \\ \rho_{k,c,t}^+ &= \sum_{\mathbf{y} \in \mathcal{D}^K} \mathbf{1}\{y_k = c\} q_t^+(\mathbf{y}). \end{aligned} \tag{3.11}$$

Here, $b_{c,t}^+ = \Pr(X_{c,t} = 1 \mid \mathcal{I}_t^+)$ and $\rho_{k,c,t}^+ = \Pr(Y_{k,t} = c \mid \mathcal{I}_t^+)$. Under the target-factorized prediction, motion propagation gives

$$\begin{aligned} \boldsymbol{\rho}_{k,t+1}^- &= M_{k,t}^\top \boldsymbol{\rho}_{k,t}^+, \\ b_{c,t+1}^- &= 1 - \prod_{k \in \mathcal{K}} \left(1 - \rho_{k,c,t+1}^-\right), \end{aligned} \tag{3.12}$$

where $\boldsymbol{\rho}_{k,t}^\pm = (\rho_{k,c,t}^\pm)_{c \in \mathcal{D}}$ and $b_{c,t+1}^-$ is the predicted occupancy probability.

The assumed density occupancy posterior and its uncertainty are

$$\begin{aligned} \widetilde{q}_t(\mathbf{x}) &= \prod_{c \in \mathcal{D}} \left(b_{c,t}^+\right)^{x_c} \left(1 - b_{c,t}^+\right)^{1 - x_c}, \\ \widetilde{H}(\mathbf{b}_t^+) &= \sum_{c \in \mathcal{D}} h(b_{c,t}^+), \\ h(p) &= -p \log p - (1-p) \log(1-p). \end{aligned} \tag{3.13}$$

Here, $\mathbf{x} = (x_c)_{c \in \mathcal{D}} \in \{0,1\}^{G^2}$, $\mathbf{b}_t^+ = (b_{c,t}^+)_{c \in \mathcal{D}}$, and h is binary entropy with the convention $0 \log 0 = 0$. Equation (3.13) is the maximum-entropy product closure of its supplied cell marginals, providing a tractable control state while omitting cross-cell dependence from the exact reference belief.

For one cell, the factorized Bayes update is additive in log-odds as follows

$$\begin{aligned} \operatorname{logit}(p) &= \log \frac{p}{1-p}, & \operatorname{sigmoid}(x) &= \frac{1}{1 + \exp(-x)}, \\ \Delta\ell(z) &= z \log \frac{p_D}{p_{FA}} + (1-z) \log \frac{1-p_D}{1-p_{FA}}, & & \\ \ell_{c,t}^- &= \operatorname{logit}(b_{c,t}^-), & \ell_{c,t}^+ &= \ell_{c,t}^- + \Delta\ell(Z_{i,c,t}), \\ b_{c,t}^+ &= \operatorname{sigmoid}(\ell_{c,t}^+). & & \end{aligned} \tag{3.14}$$

Here, $p \in (0,1)$ and $x \in \mathbb{R}$ are scalar arguments, $\ell_{c,t}^{-}$ and $\ell_{c,t}^{+}$ are the pre- and post-measurement log-odds, and independent measurements contribute additional copies of $\Delta\ell(\cdot)$.

For a prior occupancy probability $b \in [0,1]$, the expected information gain (IG) from one measurement is

$$\begin{aligned} \mathrm{IG}(b) &= I(X; Z \mid b) \\ &= h(p_{FA} + (p_D - p_{FA})b) - b\,h(p_D) - (1-b)h(p_{FA}), \end{aligned} \tag{3.15}$$

where $X \sim \mathrm{Bernoulli}(b)$, Z follows Equation (3.8), and $I(\cdot;\cdot)$ denotes mutual information.

Proposition 1. *For the sensor in Equation* (3.8)*, the expected reduction in binary occupancy entropy equals the mutual information*

$$\mathbb{E}_Z\Big[h(b) - h(b^Z)\Big] = H(X) - H(X \mid Z) = I(X; Z \mid b) = \mathrm{IG}(b) \geq 0, \tag{3.16}$$

where $b^Z = \Pr(X = 1 \mid Z)$, H is Shannon entropy, and the expectation is under the predictive distribution of Z. The inequality is strict for $0 < b < 1$ and $p_D \neq p_{FA}$.

A cell is declared confirmed when its posterior occupancy probability reaches $p_{\mathrm{conf}} = 0.95$. The equivalent log-odds threshold is $\tau_{\mathrm{conf}} = \mathrm{logit}(0.95)$. Neglecting threshold overshoot, the expected number of independent positive-state measurements is

$$\mathbb{E}[n \mid X = 1] \simeq \frac{\tau_{\mathrm{conf}} - \ell_{c,0}}{D_{\mathrm{KL}}(\mathrm{Bern}(p_D) \,\|\, \mathrm{Bern}(p_{FA}))}, \tag{3.17}$$

where X is the fixed-cell occupancy variable, n is its confirmation stopping time, $\ell_{c,0}$ is the initial log-odds, $\mathrm{Bern}(p)$ is a Bernoulli distribution, and $D_{\mathrm{KL}}(P\|Q)$ is the Kullback–Leibler divergence.

The cooperative search problem is modeled as a communication-constrained Dec-POMDP

$$\mathcal{G} = \langle \mathcal{U}, \mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{U}}, P, \{\mathcal{O}_i\}_{i\in\mathcal{U}}, O, r, \gamma, C \rangle\,. \tag{3.18}$$

Here, $\mathcal{S}$ is the state space of Equation (3.5), $\mathcal{A}_i = \mathcal{A}$ is UAV i's action space, and P is the transition kernel induced by target and UAV motion. The local observation space is $\mathcal{O}_i$, and O is the joint observation kernel induced by Equation (3.8). The team reward is r, the discount factor is $\gamma \in [0,1)$, and C is the set of communication schedules that satisfy the bandwidth constraint below.

Let $o_{i,t} \in \mathcal{O}_i$ contain UAV i's pose, energy, local coverage mask, and footprint measurements. Its execution-time information history is defined recursively by

$$\begin{aligned} \mathcal{H}_{i,0} &= (o_{i,0}, \mathcal{M}_{i,0}^{\mathrm{in}}), \\ \mathcal{H}_{i,t} &= (\mathcal{H}_{i,t-1}, a_{i,t-1}, o_{i,t}, \mathcal{M}_{i,t}^{\mathrm{in}}), \qquad t \geq 1, \end{aligned} \tag{3.19}$$

where $\mathcal{M}_{i,t}^{\mathrm{in}}$ is the set of evidence items contained in packets delivered to UAV i at step t. Decentralized actors may depend on $\mathcal{H}_{i,t}$ but not on s_t^{env} or another UAV's private history.

The atomic communication unit is a directed, provenance-tagged evidence item

$$e = (u_e, v_e, c_e, \tau_e, \iota_e, \Delta\ell_e), \qquad \Delta\ell_e = \Delta\ell(Z_{u_e,c_e,\tau_e}), \tag{3.20}$$

where $u_e, v_e \in \mathcal{U}$ are the sender and receiver, $c_e \in \mathcal{D}$ is the measured cell, $\tau_e \leq t$ is the measurement timestamp, ι_e is a globally unique provenance identifier, and $\Delta\ell_e$ is the log-likelihood innovation defined in Equation (3.14).

Let $\mathcal{E}_i(t)$ be sender i's candidate set, with $u_e = i$, $v_e \in \mathcal{U} \setminus \{i\}$, and $\tau_e = t$ for every $e \in \mathcal{E}_i(t)$. Its feasible communication schedules are

$$\begin{aligned}\mathfrak{C}_i(t) = \Bigg\{ (\boldsymbol{\chi}, \mathbf{w}) : \chi_{e,t}, w_{i,v,t} \in \{0,1\}, \\ \chi_{e,t} \le w_{i,v_e,t}, \qquad e \in \mathcal{E}_i(t), \\ \sum_{v \in \mathcal{U}\setminus\{i\}} w_{i,v,t} \le B, \\ \sum_{\substack{e \in \mathcal{E}_i(t): \\ v_e = v}} \mathrm{size}(e)\chi_{e,t} \le m w_{i,v,t}, \quad v \in \mathcal{U} \setminus \{i\} \Bigg\}.\end{aligned} \tag{3.21}$$

Here, $\boldsymbol{\chi} = (\chi_{e,t})_{e \in \mathcal{E}_i(t)}$ contains item-selection indicators, $\mathbf{w} = (w_{i,v,t})_{v \ne i}$ contains receiver-selection indicators, $\mathrm{size}(e) \in \mathbb{N}$ is the number of item slots consumed by item e, B is the maximum number of outgoing packets per UAV step, and m is the number of slots per packet. Every feasible schedule therefore satisfies

$$\sum_{e \in \mathcal{E}_i(t)} \mathrm{size}(e)\chi_{e,t} \le Q, \qquad Q = Bm, \qquad (\boldsymbol{\chi}, \mathbf{w}) \in \mathfrak{C}_i(t), \tag{3.22}$$

where Q is the aggregate item-slot budget. The implementation uses $\mathrm{size}(e) = 1$.

Exactly-once fusion is imposed by a provenance ledger. For receiver v, the newly accepted set and ledger recursion are

$$\begin{aligned}\mathcal{J}^{\mathrm{loc}}_{v,t} &= \{\iota_{v,c,t} : c \in \mathcal{F}_v(t)\}, \\ \mathcal{L}^{\mathrm{pre}}_{v,t} &= \mathcal{L}_{v,t-1} \cup \mathcal{J}^{\mathrm{loc}}_{v,t}, \\ \mathcal{A}_{v,t} &= \left\{ e \in \mathcal{M}^{\mathrm{in}}_{v,t} : v_e = v,\ \iota_e \notin \mathcal{L}^{\mathrm{pre}}_{v,t} \right\}, \\ \mathcal{L}_{v,t} &= \mathcal{L}^{\mathrm{pre}}_{v,t} \cup \{\iota_e : e \in \mathcal{A}_{v,t}\}, \qquad \mathcal{L}_{v,-1} = \varnothing.\end{aligned} \tag{3.23}$$

Here, $\iota_{v,c,t}$ is the identifier of receiver v's own measurement at cell c and step t, $\mathcal{J}^{\mathrm{loc}}_{v,t}$ is the set of such local identifiers, $\mathcal{L}^{\mathrm{pre}}_{v,t}$ is the ledger before packet fusion, and $\mathcal{L}_{v,t}$ contains all identifiers fused by receiver v through step t.

The task reward uses the same operational components evaluated in the experiments and is defined as follows:

$$r^{\mathrm{task}}_t = \lambda_D \Delta D_t + \lambda_F \Delta F_t - \lambda_R R_t - \lambda_C C_t - \lambda_E E_t. \tag{3.24}$$

Here, ΔD_t is the number of newly confirmed targets, ΔF_t is the incremental spatial balance and footprint coverage score, R_t is the revisit and invalid-motion cost, C_t is the number of transmitted item slots, E_t is the energy expenditure, and $\lambda_D, \lambda_F, \lambda_R, \lambda_C, \lambda_E \ge 0$ are their weights. The learning objective combines this task reward with potential-based belief shaping. Let

$$\begin{aligned}\xi_t &= (s^{\mathrm{env}}_t, \mathbf{b}^{+}_t), \\ \Phi(\xi_t) &= -\lambda_H \widetilde{H}(\mathbf{b}^{+}_t), \\ r^{\mathrm{team}}_t &= r^{\mathrm{task}}_t + \gamma \Phi(\xi_{t+1}) - \Phi(\xi_t),\end{aligned} \tag{3.25}$$

where $\mathbf{b}^{+}_t$ is the centralized occupancy marginal and $\lambda_H \ge 0$ is the uncertainty weight. The terminal state is absorbing and its potential is set to zero.

Information credit is allocated without additional environment rollouts. The likelihood batch contributed by UAV i at step t is

$$\Lambda_{i,t}(\mathbf{y}) = \prod_{c \in \mathcal{F}_i(t)} L(Z_{i,c,t} \mid x_c(\mathbf{y})). \tag{3.26}$$

For a uniformly sampled permutation σ_t of $\mathcal{U}$, define the sequential posteriors

$$\begin{aligned} q_t^{(0)}(\mathbf{y}) &= q_t^-(\mathbf{y}), \\ q_t^{(j)}(\mathbf{y}) &= \frac{q_t^{(j-1)}(\mathbf{y})\Lambda_{\sigma_t(j),t}(\mathbf{y})}{\sum_{\mathbf{y}' \in \mathcal{D}^K} q_t^{(j-1)}(\mathbf{y}')\Lambda_{\sigma_t(j),t}(\mathbf{y}')}, \quad j = 1, \ldots, N. \end{aligned} \tag{3.27}$$

Here, $\sigma_t(j)$ is the UAV in position j of the permutation. With $H(q) = -\sum_{\mathbf{y} \in \mathcal{D}^K} q(\mathbf{y}) \log q(\mathbf{y})$, its marginal information contribution is

$$G_{\sigma_t(j),t} = H(q_t^{(j-1)}) - H(q_t^{(j)}), \tag{3.28}$$

and commutativity of likelihood multiplication yields

$$\sum_{i \in \mathcal{U}} G_{i,t} = H(q_t^-) - H(q_t^+). \tag{3.29}$$

The operational implementation replaces $H(q)$ by the factorized entropy in Equation (3.13). The balanced per-UAV reward is

$$r_{i,t} = \frac{r_t^{\text{team}}}{N} + \lambda_A \left(G_{i,t} - \frac{1}{N} \sum_{j \in \mathcal{U}} G_{j,t} \right), \qquad \sum_{i \in \mathcal{U}} r_{i,t} = r_t^{\text{team}}, \tag{3.30}$$

where $\lambda_A \geq 0$ is the information credit coefficient. Therefore the cooperative objective can be written as

$$\begin{aligned} J(\Theta) &= \sum_{i \in \mathcal{U}} J_i(\Theta) \\ &= \mathbb{E}_{\pi_\Theta, P}\left[\sum_{t=0}^{\infty} \gamma^t \sum_{i \in \mathcal{U}} r_{i,t} \right] = \mathbb{E}_{\pi_\Theta, P}\left[\sum_{t=0}^{\infty} \gamma^t r_t^{\text{team}} \right], \end{aligned} \tag{3.31}$$

where J_i is UAV i's discounted return, π_Θ is the joint policy with parameter set Θ, and the expectation is over policy and environment transitions.

4. The HOCCA framework

Figure 1 illustrates the three-stage decentralized execution pipeline of HOCCA. Local sensing and accepted evidence first produce a timestamp-consistent information state. Receiver-aware selection and message-restricted graph attention then construct a coordination embedding under the hard budget in Equation (3.21). Finally, option and intraoption policies generate persistent options and primitive actions from locally available information. The global state enters only the training critics.

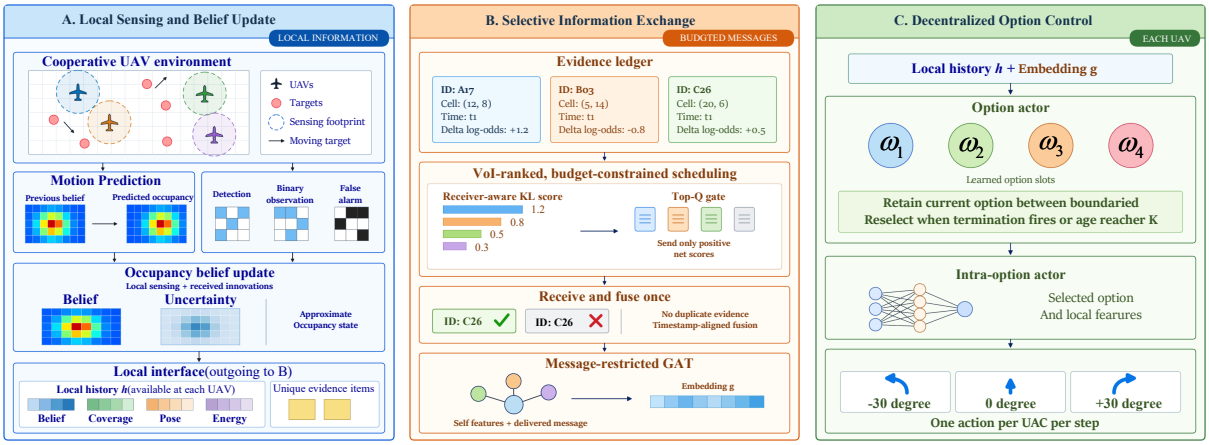


Figure 1. Architecture and information flow of HOCCA. Blue (A): Local sensing, motion prediction, and uncertainty-aware occupancy belief updating. Orange (B): Receiver-aware evidence prioritization, budget-constrained transmission, duplicate-safe fusion, and message-restricted graph attention. Green (C): Decentralized option selection, learned termination, and intraoption execution over the primitive action set $\{-30°, 0°, +30°\}$. Centralized state information is used only by the training critics and is unavailable during decentralized execution.

4.1. Provenance-aware local belief estimation

Decentralized target search couples spatially limited, error-prone sensing with target mobility and communication latency. Directly accumulating local and received detections can therefore propagate stale evidence or count the same observation more than once. To obtain a control-ready local state under these conditions, each UAV maintains a provenance-aware occupancy filter that predicts target motion, aligns innovations by measurement time, and preserves posterior uncertainty.

When target identities are not observable, UAV i propagates its occupancy map with the representative transition kernel

$$\begin{aligned} \overline{M}_t &= \frac{1}{K}\sum_{k\in\mathcal{K}} M_{k,t}, \\ \mathbf{b}_{i,r}^{-,(t)} &= \Pi_\epsilon\Big(\overline{M}_{r-1}^{\top}\mathbf{b}_{i,r-1}^{+,(t)}\Big), \qquad r \geq 1, \\ [\Pi_\epsilon(\mathbf{x})]_c &= \min\{1-\epsilon, \max\{\epsilon, x_c\}\}. \end{aligned} \tag{4.1}$$

Here, $\mathbf{b}_{i,r}^{\pm,(t)}$ is UAV i's occupancy map at measurement time r after processing all evidence received through the current step t. Here, x_c is component c of $\mathbf{x}$, $\overline{M}_t$ is row-stochastic, and $\epsilon \in (0, 1/2)$ prevents infinite log-odds. We use $\epsilon = 10^{-6}$. The common prior initializes $\mathbf{b}_{i,0}^{-,(0)}$.

The innovations from local measurements and accepted messages at the measurement time r are

$$\begin{aligned} \Delta\ell_{i,c,r}^{\mathrm{loc}} &= \begin{cases} \Delta\ell(Z_{i,c,r}), & c \in \mathcal{F}_i(r), \\ 0, & c \notin \mathcal{F}_i(r), \end{cases} \\ \Delta\ell_{i,c,r|t}^{\mathrm{msg}} &= \sum_{v=0}^{t}\sum_{e\in\mathcal{A}_{i,v}} \mathbf{1}\{c_e = c, \tau_e = r\}\Delta\ell_e. \end{aligned} \tag{4.2}$$

Here, $\Delta\ell^{\mathrm{loc}}_{i,c,r} = 0$ outside the footprint, and $\mathcal{A}_{i,\nu}$ is the duplicate-free set from Equation (3.23) accepted at arrival step ν.

If a newly accepted item has the timestamp $\tau_e < t$, the stored recursion is replayed from the earliest such timestamp through t. For every replayed r, the update is

$$\begin{aligned} \ell^{+,(t)}_{i,c,r} &= \mathrm{logit}\Big(b^{-,(t)}_{i,c,r}\Big) + \Delta\ell^{\mathrm{loc}}_{i,c,r} + \Delta\ell^{\mathrm{msg}}_{i,c,r|t}, \\ b^{+,(t)}_{i,c,r} &= \mathrm{sigmoid}\Big(\ell^{+,(t)}_{i,c,r}\Big), \qquad c \in \mathcal{D}. \end{aligned} \tag{4.3}$$

Here, $\ell^{+,(t)}_{i,c,r}$ is the replayed posterior log-odds for cell c at measurement time r. Thus, the current belief is $\mathbf{b}^{+}_{i,t} = \mathbf{b}^{+,(t)}_{i,t}$, and each provenance identifier contributes to Equation (4.3) at most once. The replay state comprises message innovations indexed by (c, τ_e) and the provenance ledger. A late item with the timestamp $\tau_e < t$ triggers only the contiguous replay interval $[\tau_e, t]$, at a cost of $O((t - \tau_e)G^2)$ for a $G \times G$ occupancy map.

The posterior is combined with local coverage, pose, and energy to form the information state used by the decentralized controller

$$\begin{aligned} u_{i,c,t} &= h(b^{+}_{i,c,t}), \\ h_{i,t} &= \mathbf{b}^{+}_{i,t} \,\|\, \mathbf{u}_{i,t} \,\|\, \mathbf{v}_{i,t} \,\|\, \widetilde{\mathbf{p}}_{i,t} \,\|\, \sin\psi_{i,t} \,\|\, \cos\psi_{i,t} \,\|\, \widetilde{e}_{i,t}. \end{aligned} \tag{4.4}$$

Here, $h_{i,t}$ is the local information state, $\mathbf{u}_{i,t} = (u_{i,c,t})_{c\in\mathcal{D}}$ is the uncertainty map, $\mathbf{v}_{i,t}$ is UAV i's local coverage mask, $\widetilde{\mathbf{p}}_{i,t}$ and $\widetilde{e}_{i,t}$ are normalized position and energy, and $\|$ denotes feature concatenation.

4.2. Receiver-aware selective information exchange

Because a globally informative observation may be redundant to a particular receiver, we evaluate every directed item against the latest acknowledged receiver belief. Let $\widehat{\ell}_{i\to v,c,t}$ be sender i's time-aligned, most recently acknowledged estimate of receiver v's log-odds for cell c, initialized from the common prior. Acknowledgment overhead is included in the item size size(e). For the candidate $e \in \mathcal{E}_i(t)$ with $v_e = v$ and $c_e = c$, define

$$\begin{aligned} \widehat{b}_e &= \mathrm{sigmoid}(\widehat{\ell}_{i\to v,c,t}), \\ \widehat{b}^{+}_e &= \mathrm{sigmoid}(\widehat{\ell}_{i\to v,c,t} + \Delta\ell_e), \\ s_e &= D_{\mathrm{KL}}\Big(\mathrm{Bern}(\widehat{b}^{+}_e) \,\|\, \mathrm{Bern}(\widehat{b}_e)\Big). \end{aligned} \tag{4.5}$$

Here, $\widehat{b}_e$ and $\widehat{b}^{+}_e$ are the acknowledged receiver belief before and after item e, and $s_e \geq 0$ is its realized Bayesian surprise. Before observing the binary outcome, its expectation is

$$\mathbb{E}_{Z_e|\widehat{b}_e}[s_e] = I(X_{c_e}; Z_e \mid \widehat{b}_e) = \mathrm{IG}(\widehat{b}_e), \tag{4.6}$$

where $X_{c_e} = X_{c_e,\tau_e}$, $Z_e = Z_{u_e,c_e,\tau_e}$ follows Equation (3.8) under prior $\widehat{b}_e$, and the information gain IG is defined in Equation (3.15).

The budgeted scheduler maximizes net information value regardless of the sign of the log-likelihood innovation

$$\Big(\boldsymbol{\chi}^{\star}_{i,t}, \mathbf{w}^{\star}_{i,t}\Big) \in \arg\max_{(\boldsymbol{\chi},\mathbf{w})\in\mathfrak{C}_i(t)} \sum_{e\in\mathcal{E}_i(t)} \chi_{e,t}\,(s_e - \lambda_{\mathrm{bit}}\,\mathrm{size}(e)). \tag{4.7}$$

Here, $\chi^{\star}_{i,t}$ and $\mathbf{w}^{\star}_{i,t}$ are the optimal item and receiver indicators, and $\lambda_{\text{bit}} \geq 0$ is the per-slot communication price, set to λ_C in the implementation. For unit-size items and fixed scores, the scheduler enumerates receiver subsets of a size of at most B, retains each receiver's m largest positive net values, and selects the subset maximizing Equation (4.7).

Only successfully delivered, newly accepted evidence is permitted to enter the coordination encoder. Its local and message features are

$$\begin{aligned} f_{i\to i,t} &= \phi_{\text{self}}(h_{i,t}), \\ f_{j\to i,t} &= \phi_{\text{msg}}(\{e \in \mathcal{A}_{i,t} : u_e = j\}), \qquad j \neq i, \\ \mathcal{N}_{i,t} &= \{i\} \cup \{j \in \mathcal{U} \setminus \{i\} : \exists e \in \mathcal{A}_{i,t},\ u_e = j\}. \end{aligned} \tag{4.8}$$

Here, ϕ_{self} is a learnable local-state encoder, ϕ_{msg} is a permutation-invariant encoder of delivered item headers and payloads, $f_{j\to i,t} \in \mathbb{R}^{d_f}$, d_f is the common feature dimension, and $\mathcal{N}_{i,t}$ is receiver i's accessible neighborhood.

For K_h attention heads, the message-restricted graph embedding is

$$\begin{aligned} z^k_{ij,t} &= \text{LeakyReLU}\Big(\mathbf{a}_k{}^{\top}\left[W_k f_{i\to i,t} \,\|\, W_k f_{j\to i,t}\right]\Big), \\ \alpha^k_{ij,t} &= \frac{\exp(z^k_{ij,t})}{\sum_{j'\in\mathcal{N}_{i,t}} \exp(z^k_{ij',t})}, \\ g_{i,t} &= \Big\|_{k=1}^{K_h} \varphi\left(\sum_{j\in\mathcal{N}_{i,t}} \alpha^k_{ij,t} W_k f_{j\to i,t}\right). \end{aligned} \tag{4.9}$$

Here, $z^k_{ij,t}$ is the head-k compatibility score, $\alpha^k_{ij,t}$ is its neighborhood-normalized attention coefficient, $W_k \in \mathbb{R}^{d_h\times d_f}$ and $\mathbf{a}_k \in \mathbb{R}^{2d_h}$ are the parameters of head k, φ is an elementwise activation, d_h is the head dimension, and $g_{i,t} \in \mathbb{R}^{K_h d_h}$. We use $d_f = d_h = 16$. An undelivered packet creates neither a feature nor an attention edge.

4.3. Decentralized option-critic control

Primitive steering alone produces short-sighted switching in a long-horizon search. We therefore maintain a temporally extended option while retaining primitive action responsiveness at every step. Let Ω be the finite option set and define the local controller input

$$\zeta_{i,t} = h_{i,t} \,\|\, g_{i,t}. \tag{4.10}$$

The option policy, intraoption policy, and learned termination function are

$$\mu_i(\omega \mid \zeta_{i,t}), \qquad \pi_{i,\omega}(a \mid \zeta_{i,t}), \qquad \beta_{i,\omega}(\zeta_{i,t}) \in [0,1], \tag{4.11}$$

where $\omega \in \Omega$, $a \in \mathcal{A}$, and all three functions are parameterized locally at UAV i.

Let $\kappa_{i,t} \in \{0,\ldots,K_L-1\}$ be the age of the active option before executing step t, with $\kappa_{i,0} = 0$. Primitive execution and effective termination are

$$\begin{aligned} a_{i,t} &\sim \pi_{i,\omega_{i,t}}(\cdot \mid \zeta_{i,t}), \\ \overline{\beta}_{i,t+1} &= 1 - \left[1 - \beta_{i,\omega_{i,t}}(\zeta_{i,t+1})\right]\mathbf{1}\{\kappa_{i,t} < K_L - 1\}, \\ d^{\text{opt}}_{i,t+1} &\sim \text{Bernoulli}(\overline{\beta}_{i,t+1}). \end{aligned} \tag{4.12}$$

Here, $K_L \in \mathbb{N}$ is the maximum option duration, $\overline{\beta}_{i,t+1}$ is the effective termination probability, and $d^{\text{opt}}_{i,t+1} = 1$ denotes an option boundary. The option and age recursions are

$$\begin{aligned} \omega_{i,t+1} &\sim \begin{cases} \mu_i(\cdot \mid \zeta_{i,t+1}), & d^{\text{opt}}_{i,t+1} = 1, \\ \delta_{\omega_{i,t}}, & d^{\text{opt}}_{i,t+1} = 0, \end{cases} \\ \kappa_{i,t+1} &= (1 - d^{\text{opt}}_{i,t+1})(\kappa_{i,t} + 1), \end{aligned} \tag{4.13}$$

where $\delta_{\omega_{i,t}}$ is a point mass at the current option and $\omega_{i,0} \sim \mu_i(\cdot \mid \zeta_{i,0})$.

During centralized training, let s_t contain s^{env}_t and all belief-filter, ledger, and option-age variables required to make the critic transition Markov. Let $\boldsymbol{\omega}_t = (\omega_{1,t}, \ldots, \omega_{N,t})$ and $\mathbf{a}_t = (a_{1,t}, \ldots, a_{N,t})$. The agent-indexed action and option critics are

$$\begin{aligned} Q_{U,i}(s_t, \boldsymbol{\omega}_t, \mathbf{a}_t) &= \mathbb{E}\big[r_{i,t} + \gamma(1 - d^{\text{env}}_t) U_i(s_{t+1}, \boldsymbol{\omega}_t) \,\big|\, s_t, \boldsymbol{\omega}_t, \mathbf{a}_t\big], \\ Q_{\Omega,i}(s_t, \boldsymbol{\omega}_t) &= \sum_{\mathbf{a} \in \mathcal{A}^N} \left[\prod_{j \in \mathcal{U}} \pi_{j,\omega_{j,t}}(a_j \mid \zeta_{j,t}) \right] Q_{U,i}(s_t, \boldsymbol{\omega}_t, \mathbf{a}). \end{aligned} \tag{4.14}$$

Here, $d^{\text{env}}_t \in \{0, 1\}$ indicates the episode's termination, and U_i is the continuation value defined below.

For any candidate next option $\omega'_j \in \Omega$, define

$$p_{j,t+1}(\omega'_j) = (1 - \overline{\beta}_{j,t+1}) \mathbf{1}\{\omega'_j = \omega_{j,t}\} + \overline{\beta}_{j,t+1} \mu_j(\omega'_j \mid \zeta_{j,t+1}). \tag{4.15}$$

Because termination and reselection are conditionally independent across UAVs, given their next local inputs, the asynchronous joint continuation value is

$$U_i(s_{t+1}, \boldsymbol{\omega}_t) = \sum_{\boldsymbol{\omega}' \in \Omega^N} \left[\prod_{j \in \mathcal{U}} p_{j,t+1}(\omega'_j) \right] Q_{\Omega,i}(s_{t+1}, \boldsymbol{\omega}'), \tag{4.16}$$

where $\boldsymbol{\omega}' = (\omega'_1, \ldots, \omega'_N)$. Hence, only UAVs whose termination variables fire will resample their options.

The temporal difference target and centralized critic loss are

$$\begin{aligned} y_{i,t} &= r_{i,t} + \gamma(1 - d^{\text{env}}_t) U_i(s_{t+1}, \boldsymbol{\omega}_t), \\ \mathcal{L}_Q &= \frac{1}{N} \sum_{i \in \mathcal{U}} \mathbb{E}\Big[\big(y_{i,t} - Q_{U,i}(s_t, \boldsymbol{\omega}_t, \mathbf{a}_t)\big)^2\Big]. \end{aligned} \tag{4.17}$$

The intraoption actor marginalizes only UAV i's primitive action

$$\begin{aligned} A^{\text{act}}_{i,t} &= Q_{U,i}(s_t, \boldsymbol{\omega}_t, \mathbf{a}_t) \\ &\quad - \sum_{\bar{a} \in \mathcal{A}} \pi_{i,\omega_{i,t}}(\bar{a} \mid \zeta_{i,t}) Q_{U,i}\big(s_t, \boldsymbol{\omega}_t, (\bar{a}, \mathbf{a}_{-i,t})\big), \\ \nabla_{\theta_{\pi_i}} J_i &= \mathbb{E}\Big[\nabla_{\theta_{\pi_i}} \log \pi_{i,\omega_{i,t}}(a_{i,t} \mid \zeta_{i,t}) A^{\text{act}}_{i,t}\Big]. \end{aligned} \tag{4.18}$$

Here, $\mathbf{a}_{-i,t}$ contains all actions except $a_{i,t}$ and θ_{π_i} parameterizes UAV i's intraoption policies.

At an option boundary, the option actor uses the counterfactual advantage

$$\begin{aligned} A_{i,t}^{\mathrm{opt}} &= Q_{\Omega,i}(s_t, \boldsymbol{\omega}_t) \\ &\quad - \sum_{\bar{\omega}\in\Omega} \mu_i(\bar{\omega} \mid \zeta_{i,t}) Q_{\Omega,i}(s_t, (\bar{\omega}, \boldsymbol{\omega}_{-i,t}))\,, \\ \nabla_{\theta_{\mu_i}} J_i &= \mathbb{E}\Big[d_{i,t}^{\mathrm{opt}} \nabla_{\theta_{\mu_i}} \log \mu_i(\omega_{i,t} \mid \zeta_{i,t}) A_{i,t}^{\mathrm{opt}} \Big], \end{aligned} \tag{4.19}$$

where $d_{i,0}^{\mathrm{opt}} = 1$, $\boldsymbol{\omega}_{-i,t}$ contains the other UAVs' options, and θ_{μ_i} parameterizes μ_i.

Finally, learned termination compares continuation with local reselection as follows:

$$\begin{aligned} A_{i,t+1}^{\mathrm{term}} &= Q_{\Omega,i}(s_{t+1}, \boldsymbol{\omega}_t) \\ &\quad - \sum_{\bar{\omega}\in\Omega} \mu_i(\bar{\omega} \mid \zeta_{i,t+1}) Q_{\Omega,i}(s_{t+1}, (\bar{\omega}, \boldsymbol{\omega}_{-i,t}))\,, \\ \nabla_{\theta_{\beta_i}} J_i &= -\mathbb{E}\Big[\mathbf{1}\{\kappa_{i,t} < K_L - 1\} \nabla_{\theta_{\beta_i}} \beta_{i,\omega_{i,t}}(\zeta_{i,t+1}) \Big(A_{i,t+1}^{\mathrm{term}} + \eta_{\mathrm{delib}} \Big) \Big]. \end{aligned} \tag{4.20}$$

Here, θ_{β_i} parameterizes the termination function and $\eta_{\mathrm{delib}} > 0$ is the deliberation cost. The indicator excludes forced boundaries from the learned termination gradient.

4.4. Computational complexity

At execution time, each UAV performs belief maintenance, budgeted evidence selection, message-restricted attention, and local option control; centralized critics are inactive. Table 2 gives the dominant costs. The scheduler enumerates at most $\sum_{j=0}^{B} \binom{N-1}{j}$ receiver subsets and transmits at most $Q = Bm$ item slots. Exact training marginalizations contain $|\mathcal{A}|^N$ action terms and $|\Omega|^N$ option terms, yielding $O(|\mathcal{A}|^N |\Omega|^N)$ action-critic evaluations when directly nested. Active-step working memory is $O(G^2 + |\mathcal{L}_i| + |\mathcal{E}_i(t)|)$.

Table 2. Dominant computational costs of HOCCA. Here, G is the grid side length, N is the fleet size, K_h is the number of attention heads, $d_f = d_h$ is the feature dimension, Ω is the option set, and $\mathcal{A}$ is the primitive action set. Fixed-width local encoders contribute an additional $O(G^2 + |\mathcal{A}_{i,t}|)$ cost.

Operation	Dominant cost
Belief prediction/update	$O(G^2 + \lvert\mathcal{F}_i(t)\rvert)$ per UAV-step
Delayed evidence replay	$O((t - \tau_e)G^2)$ per late item
VoI scoring	$O(\lvert\mathcal{E}_i(t)\rvert)$
Budgeted selection	$O\Big(\sum_{j=0}^{B} \binom{N-1}{j} \lvert\mathcal{E}_i(t)\rvert \log \lvert\mathcal{E}_i(t)\rvert\Big)$
Graph attention	$O(\lvert\mathcal{N}_{i,t}\rvert K_h d_f d_h)$
Option/action inference	$O(G^2 + d_f K_h d_h + \lvert\Omega\rvert + \lvert\mathcal{A}\rvert)$
Training marginalizations	$O(\lvert\mathcal{A}\rvert^N)$ action terms and $O(\lvert\Omega\rvert^N)$ option terms
Nested critic target	$O(\lvert\mathcal{A}\rvert^N \lvert\Omega\rvert^N)$ direct action-critic evaluations

Exact critic targets are evaluated by reusing the joint policy–probability products and processing joint action–option tuples in chunks of $K_Q = 2048$. Each distinct action critic value is reused across the

option and termination expectations. Writing $A = |\mathcal{A}|$, $L = |\Omega|$, and $C_Q(N)$ for one agent-indexed critic evaluation, the resulting cost is

$$C_{\text{target}}(N) = O\Big((AL)^N C_Q(N)\Big), \qquad M_{\text{chunk}} = O(K_Q d_Q), \tag{4.21}$$

where d_Q is the temporary storage per evaluated tuple. Chunking controls the working memory while retaining every term in the expectation. For $A = 3$ and $L = 4$, the number of tuples per critic and transition is 20 736 at $N = 4$ and 429 981 696 at $N = 8$. Training is performed offline; execution adds one local controller per UAV, with at most $Bm = 16$ outgoing evidence slots per UAV step in the reference configuration.

5. Experiments

5.1. Experimental setup

The reference environment is a 48×48 grid (2304 cells) with $K = 16$ targets: 12 static and four moving. At each hop, a moving target remains in place with probability $1 - p_m$ and otherwise follows a valid eight-neighbor transition; $T_m = 20$ and $p_m = 0.25$. The default fleet has $N = 4$ UAVs, unit speed, sensing radius $R_s = 3$, detection probability $p_D = 0.90$, and false-alarm probability $p_{FA} = 0.02$. Each UAV can send at most $B = 2$ messages per step, with at most $m = 8$ items of evidence per message. Since all-to-all transmission requires $N-1 = 3$ outgoing messages per UAV step, the reference budget is binding. The operational filter uses the uniform sparse target prior $b_c(0) = K/G^2 = 16/2304$. Training uses a 260-step horizon; reference evaluations, component ablations, and communication budget evaluations use 400 steps. Reference-trained policies are evaluated without retraining for changes in target density and map scale; fleet-size experiments retrain at each N with a 400-step evaluation horizon. Map-scale evaluations use $H_{\text{eval}}(G) = \lceil 400G/48 \rceil$; target density evaluations retain a 260-step deadline. Reference coverage, recall, precision, and movement cost are reported at $t_{\text{rep}} = 260$, whereas threshold times use the full evaluation horizon. All evaluations reuse the same paired map, target initialization, and sensor realization across methods for a given seed and map index.

The reward implementation uses coefficients $(\lambda_H, \lambda_D, \lambda_R, \lambda_C, \lambda_E, \lambda_F) = (1, 25, 1.5, 0.3, 0.02, 0.2)$ for entropy, confirmed detections, revisits, communication, energy, and spatial balance terms, respectively. Policies are trained for 500 episodes with three independent seeds. Experiments use one NVIDIA RTX 4090 graphics processing unit (GPU). Table 3 lists the remaining hyperparameters.

Table 3. Hyperparameters shared across methods where applicable. GAT denotes graph attention network, MAPPO denotes multiagent proximal policy optimization, and GAE denotes generalized advantage estimation.

Parameter	Value
Grid/targets (static, moving)	48×48 / $(12, 4)$
Sensor radius/p_D/p_{FA}	3 / 0.90 / 0.02
Actions/heading resolution	$\{-30°, 0°, +30°\}$ / 12
Target hop T_m/mobility p_m	20 / 0.25
Message budget B/payload m	2 / 8 items
Training/reference evaluation horizon	260 / 400
Training episodes	500
Discount γ/replay/batch	0.99 / 3×10^4 / 128
Learning rates (critic/actor)	10^{-3} / 5×10^{-4}
Target-network smoothing τ_Q/final ε	0.005 / 0.05
Confirmation posterior p_{conf}	0.95
Options $\lvert\Omega\rvert$/maximum age K_L	4 / 8
GAT heads K_h/node dimension	2 / 16
MAPPO (clip/GAE λ/epochs)	0.2 / 0.95 / 4
Seeds/evaluation maps per seed	3 / 30

5.2. Implementation details

Both the self-encoder and controller receive the full information state $h_{i,t}$ in Equation (4.4). Their shared deterministic preprocessing mean-pools its three maps to 7×7 and concatenates the normalized position, heading, and energy, giving 152 internal features. The self-encoder has the widths 152–64–16. The message encoder applies a 6–32–16 multilayer perceptron to the normalized cell coordinates, observation age, likelihood innovation, sender index, and receiver index, followed by mean pooling over accepted items. The two 16-dimensional attention heads use bias-free projections in Equation (4.9). The controller concatenates the 152 local inputs with the 32-dimensional graph embedding and uses a shared 184–128–64 torso. Its output heads contain four option logits, four termination probabilities, and twelve intra-option action logits.

Adam updates use the rates in Table 3, gradient-norm clipping at 10, a 1,000-transition replay warm-up, and one update per environment step. Option selection is updated at option boundaries, intraoption policies at every sampled transition, and termination heads on nonforced primitive transitions, following Equations (4.18)–(4.20). Target networks use $\tau_Q = 0.005$, and primitive exploration decreases linearly from 1 to 0.05 over the first 200 episodes. Actor–critic methods use the four validation learning rate pairs in $\{5 \times 10^{-4}, 10^{-3}\} \times \{2.5 \times 10^{-4}, 5 \times 10^{-4}\}$, with 500 episodes per configuration and 10 common validation maps.

5.3. Baselines

Five baseline methods are compared. Independent Q-learning (IQL) uses a separate double DQN per UAV [37, 38]. QMIX learns a monotone factorization of the joint action value [8], and MADDPG uses per-agent actors and centralized critics [6]. MAPPO uses clipped proximal policy optimization

(PPO) with a centralized value function [7]. Our implementation of DTH-MADDPG follows the design in [2]. A Voronoi allocator is updated every 30 steps and provides goals to primitive controllers. All methods receive the same local belief window representation, sensor realizations, action constraints, and task reward components. All methods share the episode budget, evaluation maps, and hardware in Table 3, with replay, batch, and optimizer settings matched where applicable; validation-selected hyperparameters are fixed before held-out testing.

5.4. Metrics

Search effectiveness is evaluated using cumulative footprint coverage and confirmed-target recall

$$\begin{aligned} C(t) &= \frac{1}{G^2}\left|\bigcup_{\tau=0}^{t}\bigcup_{i=1}^{N}\mathcal{F}_i(\tau)\right|, \\ R_{\mathrm{det}}(t) &= \frac{K_{\mathrm{conf}}(t)}{K}. \end{aligned} \tag{5.1}$$

Here, G^2 is the number of grid cells, N is the number of UAVs, $\mathcal{F}_i(\tau)$ is the sensing footprint of UAV i at step τ, and $|\cdot|$ denotes set cardinality. K is the total number of targets, and $K_{\mathrm{conf}}(t)$ counts distinct ground truth targets confirmed by step t under the prescribed confirmation criterion. The evaluator matches a confirmed cell to a target's location at the confirmation step and retains that target's confirmation thereafter, without exposing the ground truth to the policies; repeated sensing or confirmation leaves the respective counts unchanged.

Detection precision at the reporting time measures the reliability of target confirmations

$$P_{\mathrm{det}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{5.2}$$

Here, TP and FP are the numbers of true and false target confirmations accumulated during the reporting interval, respectively. For episodes ending earlier, cumulative metrics retain their terminal values.

Search efficiency is measured by the first-passage times to 80% coverage and recall

$$\begin{aligned} T_{80}^{\mathrm{cov}} &= \min\left\{H_{\mathrm{eval}}, \inf\left\{t \in \{0,\dots,T_{\mathrm{end}}\} : C(t) \geq 0.8\right\}\right\}, \\ T_{80}^{\mathrm{det}} &= \min\left\{H_{\mathrm{eval}}, \inf\left\{t \in \{0,\dots,T_{\mathrm{end}}\} : R_{\mathrm{det}}(t) \geq 0.8\right\}\right\}. \end{aligned} \tag{5.3}$$

Here, H_{eval} is the prescribed evaluation horizon, $T_{\mathrm{end}} \leq H_{\mathrm{eval}}$ is the observed episode length, and $\inf \varnothing = +\infty$; episodes not reaching a threshold contribute H_{eval} to its horizon-capped mean and paired test. Lower values indicate faster search.

Mission success rate and complete-search time quantify whether and how quickly all targets are confirmed

$$\begin{aligned} \mathrm{SR} &= \frac{E_{\mathrm{succ}}}{E}, \\ T_{\mathrm{all}} &= \frac{1}{E_{\mathrm{succ}}}\sum_{e\in\mathcal{E}_{\mathrm{succ}}} \min\left\{t \in \{0,\dots,T_{\mathrm{end}}^{(e)}\} : K_{\mathrm{conf}}^{(e)}(t) = K\right\}. \end{aligned} \tag{5.4}$$

Here, E is the number of evaluation episodes, $\mathcal{E}_{\mathrm{succ}}$ is the set of episodes that confirm all targets within the prescribed horizon, and $E_{\mathrm{succ}} = |\mathcal{E}_{\mathrm{succ}}|$. Superscript (e) denotes an episode-specific quantity.

5.5. *Training Dynamics*

To evaluate the learning efficiency, convergence quality, and training stability of the compared methods, all agents are trained under the same environment and a common budget of 500 episodes, with the results aggregated over three independent random seeds. Figure 2 reports the smoothed mean episode reward, where the shaded region represents one standard deviation around the mean. The brown, purple, red, green, and orange curves denote HOCCA, DTH-MADDPG, MAPPO, MADDPG, and QMIX, respectively.

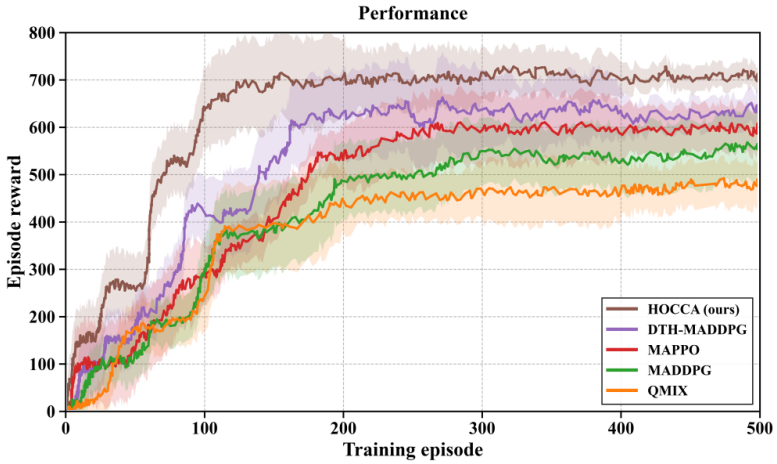


Figure 2. Measured training dynamics over 500 episodes. Solid curves show the smoothed mean episode reward across three independent seeds, and shaded regions indicate ± one standard deviation. Color coding: HOCCA (brown), DTH-MADDPG (purple), MAPPO (red), MADDPG (green), and QMIX (orange). Higher values indicate better performance.

As shown in Figure 2, HOCCA achieves both the fastest reward improvement and the highest converged return. Its mean reward rises rapidly after approximately 50 episodes, exceeds 600 near episode 100, and reaches a stable plateau of approximately 700–720 after about 150 episodes. At the end of training, HOCCA attains a mean return of approximately 710, compared with about 640 for DTH-MADDPG, 595 for MAPPO, 550–560 for MADDPG, and 480 for QMIX. The persistent advantage over the entire middle and late stages demonstrates the superior sample efficiency and asymptotic performance of the proposed method.

Among the baselines, DTH-MADDPG provides the strongest performance and eventually stabilizes around 620–650, whereas MAPPO converges more gradually to approximately 600. MADDPG and QMIX exhibit similar early-stage progress, but MADDPG establishes a clear advantage after roughly 180 episodes and converges near 550, while QMIX remains below 500. The wider uncertainty regions during the early and intermediate stages reflect seed-dependent exploration and transient policy changes; after approximately 200 episodes, the mean ordering HOCCA > DTH-MADDPG > MAPPO > MADDPG > QMIX remains stable, and the uncertainty bands generally contract as the policies converge.

5.6. *Reference environment search performance*

We evaluate the end-to-end cooperative search performance of all methods in the 48×48 reference environment using two complementary metrics. Footprint coverage measures the fraction of the

environment explored by the UAV team, whereas confirmed-target recall measures the proportion of targets reliably identified within the rollout horizon. Figure 3 presents the empirical trajectories aggregated over 90 held-out evaluations from three independently trained seeds. Solid curves denote mean performance, shaded regions represent ± one standard deviation, and the horizontal dashed line marks the operational threshold of 0.8. The gray, light-blue, orange, green, purple, and dark blue curves correspond to IQL, QMIX, MADDPG, MAPPO, DTH-MADDPG, and HOCCA, respectively.

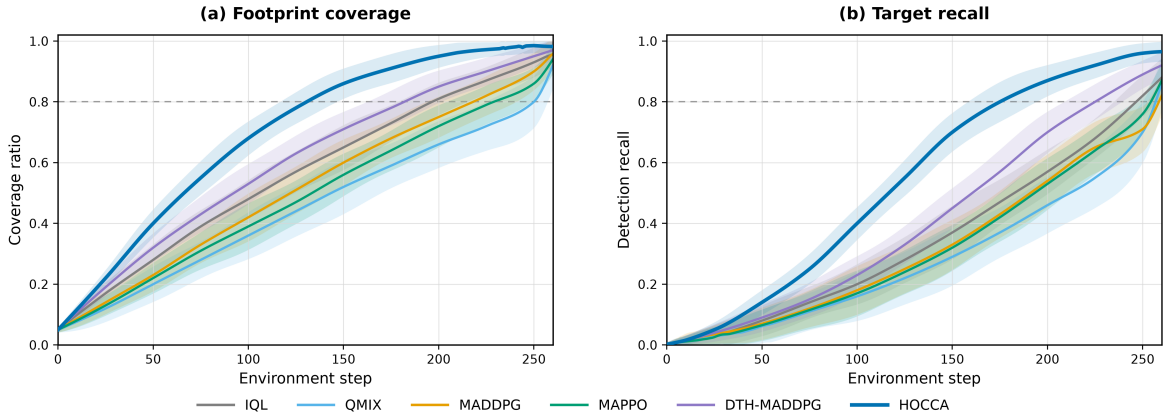


Figure 3. Measured held-out search performance in the 48 × 48 reference environment. Panel (a) reports footprint coverage, and panel (b) reports confirmed-target recall. Solid curves show the mean over 90 held-out evaluations, shaded regions indicate ± one standard deviation, and the dashed horizontal line denotes the 0.8 performance threshold. Colors identify IQL (gray), QMIX (light blue), MADDPG (orange), MAPPO (green), DTH-MADDPG (purple), and HOCCA (dark blue). Higher values and earlier threshold crossings indicate better performance.

As shown in Figure 3, HOCCA separates from the competing methods during the early stage of both search processes and maintains this advantage throughout the displayed interval. It reaches 80% footprint coverage after 137 steps, whereas the strongest baseline, DTH-MADDPG, requires 183 steps, corresponding to 25.1% fewer steps and a 1.34× speedup. For confirmed-target recall, HOCCA reaches the same threshold after 176 steps compared with 231 steps for DTH-MADDPG, reducing the required search time by 23.8% and providing a 1.31× speedup. The proposed method also reaches the highest coverage and recall at 260 steps, demonstrating that its faster progress does not sacrifice search quality.

Among the baselines, DTH-MADDPG exhibits the strongest overall trajectories, which supports the value of temporal abstraction in long-horizon multi-UAV searches. Nevertheless, its persistent separation from HOCCA suggests that hierarchical decision-making alone does not fully address the information allocation problem under uncertain sensing. IQL, QMIX, MADDPG, and MAPPO progress more slowly and exhibit considerable overlap during the early and intermediate stages, particularly for target recall. Several baselines improve rapidly near the end of the displayed interval, but these delayed gains occur too late to match the threshold-crossing efficiency of HOCCA. The broader uncertainty regions of the baseline methods also indicate greater sensitivity to the evaluation conditions, whereas the relatively compact late-stage band of HOCCA reflects more consistent cooperative behavior. The differences are quantified more precisely in Table 4.

Table 4. Main comparison on the 48×48 environment (mean ± standard deviation over three training seeds and 30 paired evaluation environments per seed; T_{80} in steps). ** denotes Holm-adjusted $p < 0.01$ for coverage time against HOCCA.

Method	T_{80}^{cov}	T_{80}^{det}	Coverage	Recall	Precision	Moves
IQL	202±15**	249±20	0.96±0.01	0.87±0.03	0.89±0.03	965±35
QMIX	247±21**	258±23	0.91±0.02	0.84±0.03	0.87±0.03	1014±45
MADDPG	216±16**	259±18	0.95±0.01	0.88±0.03	0.91±0.02	981±37
MAPPO	231±19**	248±21	0.93±0.02	0.86±0.03	0.89±0.03	998±41
DTH-MADDPG	183±14**	231±17	0.96±0.01	0.92±0.02	0.92±0.02	936±31
HOCCA (ours)	137±9	176±11	0.98±0.01	0.96±0.01	0.95±0.01	845±24

Table 5. Paired Wilcoxon tests for T_{80}^{cov} over the 90 pooled evaluation episodes (three seeds $\times$ 30 paired maps; HOCCA versus each baseline; Holm correction).

Baseline	Adjusted p-value
DTH-MADDPG	4.10×10^{-3}
MADDPG	1.20×10^{-4}
MAPPO	2.80×10^{-5}
QMIX	8.70×10^{-6}
IQL	3.50×10^{-4}

Table 4 complements the time-resolved trajectories by summarizing search efficiency, effectiveness, and motion cost. The results are reported as the mean and standard deviation over three independently trained seeds, with each seed evaluated on the same 30 paired environments. The threshold metrics T_{80}^{cov} and T_{80}^{det} record the numbers of steps required to reach 80% coverage and recall, respectively, and should therefore be minimized. Conversely, coverage, recall, and precision should be maximized.

Beyond its threshold-crossing advantage, HOCCA obtains the best coverage, recall, and precision, reaching 0.98 ± 0.01, 0.96 ± 0.01, and 0.95 ± 0.01, respectively. Compared with DTH-MADDPG, these results represent absolute improvements of 0.02 in coverage, 0.04 in recall, and 0.03 in precision. HOCCA also requires only 845 ± 24 movements, which is 9.7% fewer than DTH-MADDPG and between 12.4% and 16.7% fewer than the remaining baselines.

Table 5 examines whether the reductions in coverage time remain statistically reliable across paired evaluation environments. The tests use 90 episode pairs indexed by training seed and evaluation map, while Table 4 reports variation across the three seed-level means; the p-values characterize paired evaluation differences for these trained policies. The paired Wilcoxon test compares methods on identical environment instances without assuming normally distributed differences, while the Holm correction controls the family-wise error rate across the five baseline comparisons. All adjusted p-values are below 0.01. Even the closest comparison with DTH-MADDPG remains significant at $p = 4.10 \times 10^{-3}$, and the comparisons with the other baselines range from 8.70×10^{-6} to 3.50×10^{-4}. As a raw effect size, the mean reduction in coverage time is 46 steps versus DTH-MADDPG and ranges from 65 to 110 steps versus the other baselines. Together, the paired comparisons and seed-level summaries support faster exploration, more accurate confirmation, and lower motion cost under the shared evaluation protocol.

5.7. *Target-density and map-scale generalization*

We assess policy transfer by varying the target count $K \in \{10, 20, 30, 40\}$ and the grid side length $G \in \{32, 48, 64, 80\}$ separately, retaining the reference target motion model, sensor parameters, and communication budget, with map-scale horizons $H_{\text{eval}}(G) = \lceil 400G/48 \rceil$. The target density sweep uses the 48×48 reference grid with a 260-step deadline. Figure 4(a) reports the mission success rate, whereas Figure 4(b) reports T_{80}^{cov}; higher success rates and lower coverage times indicate better performance. Table 6 additionally reports the complete search time T_{all} conditional on successful episodes.

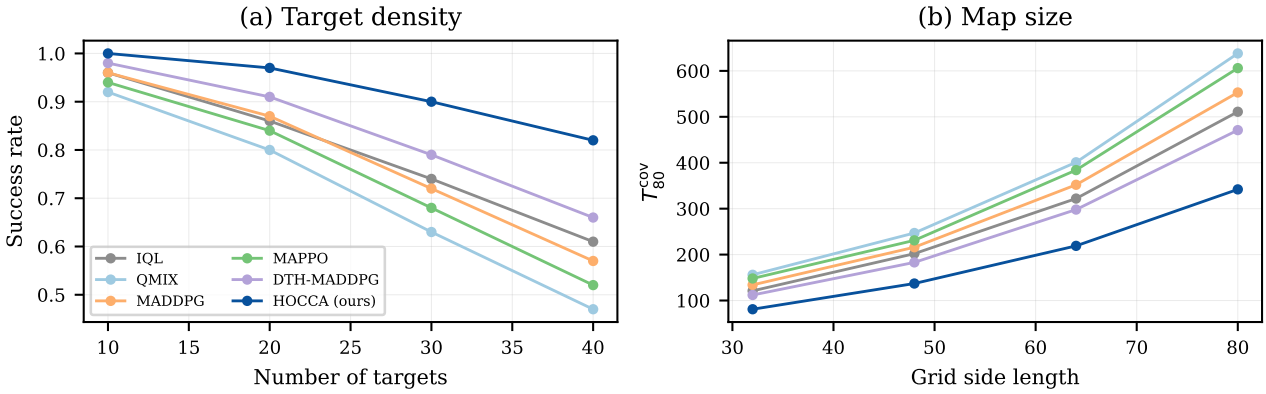


Figure 4. Measured generalization performance under increasing task complexity. Panel (a) shows mission success rate as the target count increases from 10 to 40. Panel (b) shows the number of steps required to reach 80% footprint coverage as the grid side length increases from 32 to 80. Colors identify IQL (gray), QMIX (light blue), MADDPG (orange), MAPPO (green), DTH-MADDPG (purple), and HOCCA (dark blue). In Panel (a), higher values are better. In Panel (b), lower values are better.

As the target density increases, the success rate decreases for every method, but the degradation of HOCCA is substantially slower. Its success rate remains at 1.00, 0.96, 0.89, and 0.81 for $K = 10$, 20, 30, and 40, respectively. At the most demanding setting, $K = 40$, HOCCA exceeds DTH-MADDPG and IQL by 0.16 and 0.21 in absolute success rate, respectively, and exceeds QMIX by 0.35. The conditional completion times exhibit the same advantage: HOCCA confirms all targets in 139 steps at $K = 10$, 188 steps at $K = 20$, and 228 steps at $K = 30$, compared with 181, 234, and 256 steps for DTH-MADDPG. At $K = 40$, HOCCA is the only method for which a stable T_{all} estimate is obtained, completing successful missions in an average of 249 steps. The map scale results further demonstrate that the advantage of HOCCA persists as the search space expands. Although T_{80}^{cov} increases monotonically for all methods, HOCCA remains the fastest at every tested grid size, requiring 79, 135, 217, and 340 steps for $G = 32$, 48, 64, and 80, respectively. Relative to DTH-MADDPG, these values represent reductions of 28.2%, 25.4%, 26.7%, and 27.4%, showing that the relative advantage remains stable across the complete scale range. On the largest map, HOCCA requires 128 fewer steps than DTH-MADDPG, 169 fewer than IQL, and 296 fewer than QMIX. From $G = 32$ to $G = 80$, its coverage time grows by a factor of 4.30, while the map area grows by a factor of 6.25.

Table 6. Target density and map scale evaluation. T_{all} denotes the mean number of steps required to confirm all targets, conditional on successful episodes. A conditional estimate is reported only when each training seed has at least 20 successful episodes and the pooled 95% bootstrap confidence interval half-width is below 10 steps. Otherwise, the entry is "—".

Factor	Metric	HOCCA	DTH-MADDPG	MADDPG	MAPPO	QMIX	IQL
$K = 10$	SR	1.00	0.97	0.95	0.93	0.90	0.94
	T_{all}	139	181	203	212	224	194
$K = 20$	SR	0.96	0.90	0.86	0.83	0.79	0.85
	T_{all}	188	234	251	256	259	247
$K = 30$	SR	0.89	0.78	0.71	0.67	0.62	0.73
	T_{all}	228	256	—	—	—	257
$K = 40$	SR	0.81	0.65	0.56	0.51	0.46	0.60
	T_{all}	249	—	—	—	—	—
$G = 32$	T_{80}^{cov}	79	110	132	146	154	119
$G = 48$	T_{80}^{cov}	135	181	214	229	245	200
$G = 64$	T_{80}^{cov}	217	296	350	382	399	320
$G = 80$	T_{80}^{cov}	340	468	551	604	636	509

5.8. Fleet-size scaling

To evaluate whether the learned coordination policy can effectively exploit additional sensing and mobility resources, we retrained each method with fleet sizes $N \in \{2, 4, 6, 8\}$ on the same 48×48 environment using a fixed 400-step horizon. Figure 5 reports T_{80}^{cov} and T_{80}^{det}, which denote the numbers of environment steps required to reach 80% footprint coverage and 80% confirmed-target recall, respectively; lower values indicate more efficient cooperative searches. The gray, orange, green, purple, and dark blue curves represent IQL, MADDPG, MAPPO, DTH-MADDPG, and HOCCA, respectively.

In Table 7, both metrics decrease monotonically as the fleet grows, confirming that all evaluated methods benefit from additional UAVs; however, HOCCA converts the added agents into search progress most effectively. From $N = 2$ to $N = 8$, its coverage time decreases from 249 to 89 steps, corresponding to a 64.3% reduction and a 2.80× speedup, while its detection time decreases from 312 to 118 steps, corresponding to a 62.2% reduction and a 2.64× speedup. The largest improvement occurs when the fleet grows from two to four UAVs, after which the marginal reduction gradually diminishes because spatial overlap, communication dependencies, and coordination constraints limit perfectly linear scaling. Nevertheless, HOCCA retains the lowest coverage and detection times at every fleet size.

Table 7. Fleet-size scaling on the 48×48 grid with a 400-step horizon. Each entry reports $T_{80}^{\text{cov}}/T_{80}^{\text{det}}$ in environment steps. Lower values are better, and "—" denotes a method that was not evaluated.

Fleet	HOCCA	DTH-MADDPG	MADDPG	MAPPO	IQL
$N = 2$	249/312	292/361	322/388	341/395	306/371
$N = 4$	137/176	183/231	216/259	231/248	202/249
$N = 6$	108/141	148/189	176/214	194/207	158/196
$N = 8$	89/118	121/158	147/181	168/173	127/163

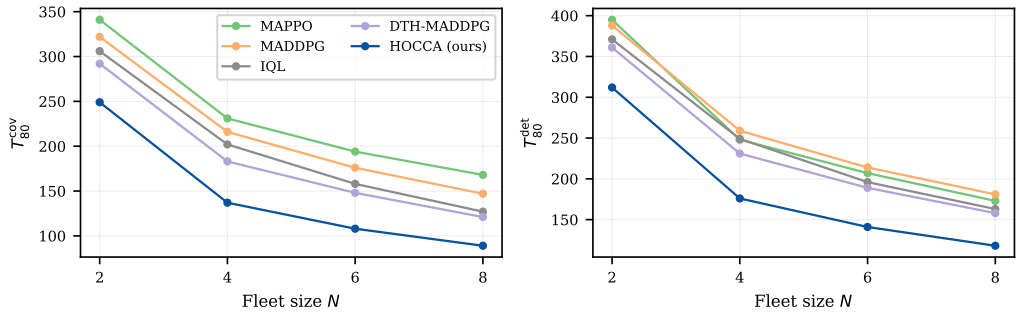


Figure 5. Measured fleet-size scaling on the 48×48 reference environment with a 400-step horizon. The two panels report the numbers of steps required to reach 80% footprint coverage and 80% confirmed-target recall as the fleet size increases from two to eight UAVs. Colors identify IQL (gray), MADDPG (orange), MAPPO (green), DTH-MADDPG (purple), and HOCCA (dark blue). Lower values indicate faster cooperative searches.

Among the baselines, DTH-MADDPG is consistently the strongest competitor, followed generally by IQL, MADDPG, and MAPPO, but none matches the scaling efficiency of HOCCA. When the fleet is quadrupled, the coverage time speedup of DTH-MADDPG is 2.41×, compared with 2.80× for HOCCA, and its detection time speedup is 2.28×, compared with 2.64× for HOCCA. At $N = 8$, HOCCA requires 89 coverage steps and 118 detection steps, whereas DTH-MADDPG requires 121 and 158 steps, meaning that HOCCA uses 26.4% fewer steps for coverage and 25.3% fewer steps for target confirmation. Its advantage is also present at $N = 2$, showing that the improvement is not solely a consequence of operating with a large fleet.

5.9. Component ablations

To identify the contribution of each architectural component, we construct five controlled variants of HOCCA by removing temporal options, the graph attention network (GAT) module, the VoI scheduler, execution-time communication, or the balanced differential reward. All variants are trained independently using the same training seeds, training budget, hyperparameters, and paired evaluation environments. Joint-removal variants compose the same component replacements used in the corresponding single-removal variants, with all remaining settings held fixed. Figure 6 summarizes the single-component comparisons, and Table 8 reports their threshold times.

Table 8. Single-component ablation results. Threshold times are reported in environment steps; lower values are better.

Variant	T_{80}^{cov}	T_{80}^{det}
Full	137	176
w/o options	189	232
w/o attention	152	196
w/o VoI gate	148	191
w/o communication	161	207
w/o differential reward	158	204

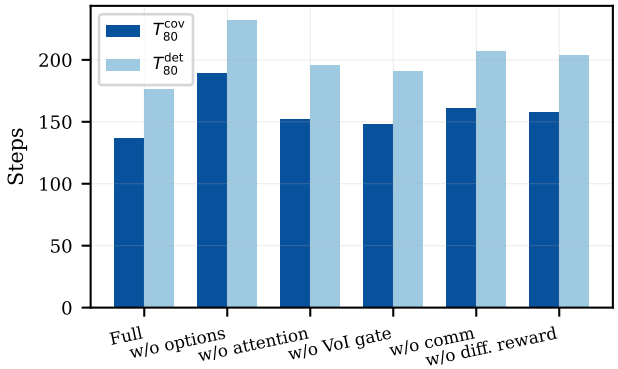


Figure 6. Single-component ablations of HOCCA. The bars report the environment steps required to reach 80% coverage and confirmed target recall. Lower values indicate faster search.

Removing options increases coverage and detection time by 38.0% and 31.8%, respectively, identifying temporal abstraction as the largest individual contributor within the complete architecture. Removing communication increases the threshold times to 161 and 207 steps, while removing differential reward yields 158 and 204 steps. Attention and VoI selection also contribute consistently: their removal increases coverage time by 10.9% and 8.0%, respectively. The variants with VoI, attention, or reward removed retain options and communication, separating evidence selection, message aggregation, and credit assignment within the same hierarchical controller. With communication disabled, the attention neighborhood contains the self node alone, preserving local feature processing while eliminating interagent information exchange.

To quantify interactions between components, we evaluate three paired removals. The pairs are options with VoI, attention with VoI, and options with differential reward. Each pair forms a 2×2 factorial comparison comprising the full model, the two corresponding single removals, and their joint removal. For each matched training seed, the interaction contrast is

$$\Delta_{u,v}^{q} = T_{-u,-v}^{q} - T_{-u}^{q} - T_{-v}^{q} + T_{\text{Full}}^{q}, \qquad q \in \{\text{cov}, \text{det}\}. \tag{5.5}$$

Here, u and v denote the components being examined, and T^{q} is the seed-level evaluation mean of T_{80}^{q}. The subscripts $-u$, $-v$, $-u,-v$, and Full identify the corresponding single removals, joint removal, and the complete model. Positive and negative contrasts indicate joint-removal penalties above and below the additive reference, respectively.

Table 9. Paired-component ablations and factorial interaction contrasts. Threshold times are reported in environment steps as the mean ± standard deviation across training seeds; interaction contrasts are reported as means computed from the corresponding ablation means. Lower threshold times indicate a faster search.

Jointly removed components	T_{80}^{cov}	T_{80}^{det}	$\Delta_{u,v}^{cov}$	$\Delta_{u,v}^{det}$
Options and VoI	189.4 ± 6.8	238.7 ± 8.5	−10.6	−8.3
Attention and VoI	171.2 ± 5.9	216.5 ± 7.2	8.2	5.5
Options and differential reward	178.6 ± 6.3	225.9 ± 7.6	−31.4	−34.1

In Table 9, joint attention–VoI removal requires 171.2 coverage steps and 216.5 detection steps, exceeding the additive references by 8.2 and 5.5 steps, respectively. The option–VoI and option–reward pairs exhibit negative contrasts, identifying overlapping, configuration-dependent removal effects. In particular, removing differential reward within the option-free controller changes the threshold times from 189/232 to 178.6/225.9 steps, showing that its contribution depends on the retained control architecture. Across all single and paired removals, the complete model retains the lowest coverage and detection times, supporting the effectiveness of the integrated controller.

5.10. Hyperparameter sensitivity and communication cost

We evaluate the sensitivity of HOCCA to four hyperparameters governing temporal abstraction, model capacity, and interagent communication. We vary the maximum option age $K_L \in \{4, 8, 16\}$, the number of options $|\Omega| \in \{2, 4, 6\}$, the number of GAT heads $K_h \in \{1, 2, 4\}$, and the communication budget $B \in \{1, 2, 3\}$. Figure 7 presents a one-factor-at-a-time analysis centered on the reference configuration $(K_L, |\Omega|, K_h, B) = (8, 4, 2, 2)$, which achieves $T_{80}^{\text{cov}} = 137$ environment steps.

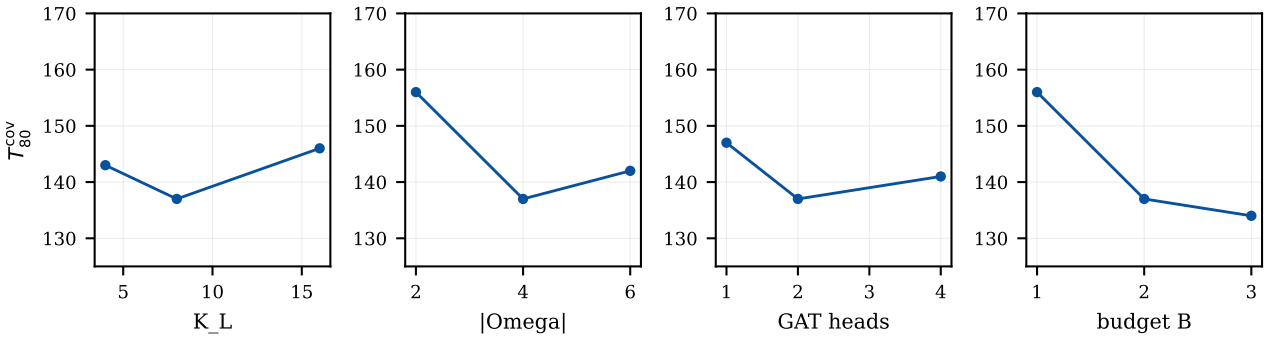


Figure 7. One-factor-at-a-time sensitivity analysis of HOCCA with respect to the maximum option age K_L, option count $|\Omega|$, number of GAT heads K_h, and communication budget B. All remaining hyperparameters are fixed at $(K_L, |\Omega|, K_h, B) = (8, 4, 2, 2)$. Markers denote the environment steps required to reach 80% footprint coverage, with lower values indicating faster exploration.

The option age sweep yields approximately 143, 137, and 146 steps for $K_L = 4$, 8, and 16, respectively, indicating that the intermediate setting balances temporal consistency against responsiveness to updated observations. The option-count sweep exhibits a more pronounced effect. Increasing $|\Omega|$ from two to four reduces the coverage time from 156 to 137 steps (12.2%), whereas increasing it further to six raises the time to 142 steps (3.6% above the reference). A similar nonmonotonic trend is observed for attention capacity, with one, two, and four GAT heads requiring approximately 147, 137, and 141 steps, respectively. Thus, two heads improve upon a single head by 6.8%, while additional heads provide no further coverage benefit.

Increasing the communication budget from $B = 1$ to $B = 2$ reduces the coverage time from approximately 156 to 137 steps (12.2%), whereas increasing it to $B = 3$ yields only a further 2.2% improvement, reaching approximately 134 steps. This modest three-step reduction requires a 50% increase in the permitted message budget, supporting $B = 2$ as a favorable trade-off between coverage efficiency and communication allowance. Across all configurations, coverage times range from 134

to 156 steps, corresponding to deviations of −2.2% to +13.9% relative to the reference. Overall, the reference settings minimize the threshold time in the option age, option count, and attention head sweeps, while a larger communication budget provides only marginal additional benefit.

5.11. Communication constraints

Table 10 compares the complete controller, its variants without options or VoI, and a TarMAC-style communication comparator under four channel impairments. The four impairments are independent 20% packet loss, a fixed two-step delay, 5-step outages every 20 steps, and bandwidth alternating between $B = 1$ and $B = 2$ in 10-step blocks. The combined condition applies loss, delay, and outages together with $B = 1$. Lost transmissions consume the attempted communication budget, acknowledgements follow the same channel, and delayed innovations enter the timestamped replay in Equation (4.3).

Table 10. Confirmed-target recall at $t = 260$ under communication constraints. Entries are means over the paired evaluation maps and training seeds.

Condition	Full	TarMAC-style	Without options	Without VoI
Nominal	0.963	0.951	0.921	0.949
20% packet loss	0.944	0.927	0.899	0.919
Two-step delay	0.940	0.921	0.894	0.912
Intermittent links	0.932	0.909	0.881	0.899
Variable bandwidth	0.948	0.929	0.905	0.922
Combined constraints	0.903	0.870	0.846	0.863

The TarMAC-style comparator uses learned query–key recipient priorities [32] with the same option controller, evidence payload, packet cap B, and item cap m. Recipient queries are generated locally from the sender's acknowledged receiver belief estimate. Data and acknowledgement bytes define the shared communication budget. It is trained with the same 500-episode budget and paired training seeds. The no-VoI variant samples eligible evidence uniformly, while preserving the same provenance-aware fusion.

The complete controller has the highest recall in every channel condition. Its advantage over the case without VoI increases from 0.014 under nominal communication to 0.040 under combined constraints; the corresponding gap against the case without options increases from 0.042 to 0.057. Receiver-specific evidence selection makes scarce deliveries informative, while temporally extended options sustain coherent local search between deliveries.

5.12. Execution requirements

Table 11 reports the reference execution profile on the RTX 4090 workstation at a batch size of one. The complete decision comprises belief maintenance, VoI scheduling, message aggregation, and option/action inference. Timing uses 100 warm-up decisions and 1,000 timed decisions with device synchronization. The deployed network has 45,396 parameters, including both encoders, graph attention, and all policy heads; its float32 weights occupy 177.3 KiB. The three 48×48 state maps occupy 27 KiB per UAV.

Table 11. Reference execution requirements for one UAV.

Quantity	Value
Actor forward-pass latency	0.19 ms
Complete local decision latency	1.12 ms
Peak allocated inference device memory	2.4 MiB
Peak host-process resident memory	326 MiB
Float32 deployed weights per UAV	177.3 KiB
Three float32 reference-state maps	27 KiB

5.13. Sensor quality and information value

To isolate the effect of sensor quality on evidence utility, we evaluate $\mathrm{IG}(b_0)$ in Equation (3.15) across detection and false-alarm probabilities, holding the occupancy prior fixed at $b_0 = 16/2304$. By Equation (4.6), this quantity also represents the expected Bayesian surprise of a measurement for a receiver with the belief b_0. Table 12 reports the resulting analytical values.

Table 12. Analytical sensitivity of per-measurement mutual information $\mathrm{IG}(b_0)$ to sensor quality at the reference occupancy prior $b_0 = 16/2304$. Values are reported in millibits (10^{-3} bits), using base-two logarithms. Bold indicates the reference operating point $(p_D, p_{FA}) = (0.90, 0.02)$.

$p_D \backslash p_{FA}$	0.01	0.02	0.05	0.10	0.20
0.80	30.0	25.4	18.7	13.5	8.3
0.90	35.9	30.8	23.3	17.4	11.4
0.95	39.3	33.9	26.0	19.7	13.3
0.99	42.3	36.7	28.5	22.0	15.3

The reference operating point supplies 30.8 millibits per measurement. Reducing p_D to 0.80 at fixed $p_{FA} = 0.02$ lowers this value to 25.4 millibits, a reduction of approximately 17.5%. Increasing p_{FA} to 0.20 at a fixed $p_D = 0.90$ reduces it to 11.4 millibits, approximately 63% below the reference. At $p_{FA} = 0.20$, raising p_D to 0.99 yields 15.3 millibits, approximately half the reference value. Thus, within the evaluated range, high detection probability alone does not preserve the information available under low false-alarm conditions.

For the sparse reference prior, false alarms reduce the diagnostic value of positive measurements because most cells are unoccupied. In HOCCA, the sensor likelihood determines each innovation in the evidence, while Equation (4.5) evaluates the realized item against the acknowledged receiver belief. Message valuation therefore accounts jointly for sensor reliability and the receiver's current information state under the prescribed communication budget.

5.14. Belief accuracy and extended validation

The occupancy approximation is assessed on a 6×6 grid with two labeled targets. The exact joint filter and operational occupancy filter receive identical evidence streams, using the reference sensor model and target-hop process. Table 13 reports the mean absolute error (MAE) of the occupancy estimates across cells and time steps $|\mathcal{D}|^{-1} \sum_c |\widehat{b}_c - b_c^\star|$ over 100 paired streams per configuration. It also

reports the dependence gap $\sum_c h(b_c^\star) - H(q^X)$ in nats, where q^X is the exact distribution over occupancy maps obtained by aggregating labeled joint states.

Table 13. Occupancy belief accuracy against the exact joint filter. MAE denotes mean absolute error.

Target configuration	Occupancy MAE	Dependence gap (nats)
Separated, static	0.008	0.12
Adjacent, static	0.017	0.29
Adjacent, moving	0.026	0.38

The occupancy mean absolute error remains between 0.008 and 0.026 across these configurations. This accuracy supports its use as the compact belief state for option control and receiver-aware evidence valuation.

Table 14 extends the principal option/VoI ablations and the $N = 8$ comparison to five independently trained checkpoints per method. Each checkpoint is averaged over 30 paired held-out maps; the reported 95% confidence interval is $\overline{T} \pm t_{0.975,4} s_T / \sqrt{5}$, with $t_{0.975,4} = 2.776$ and s_T calculated across checkpoint means. The same validation group includes one constrained scenario with 10% blocked cells and connected free space, two UAVs with $(v, R_s) = (1, 3)$ and two with $(0.8, 2)$, a temporary 4×4 no-fly patch active at Steps 100–109, and a 220-cell movement budget per UAV.

Table 14. Five-seed ablation, fleet size, and constrained scenario summaries. Entries show means and 95% confidence intervals. Threshold times are in steps, and constrained-scenario recall is reported at $t = 260$.

Setting/method	Metric	Mean [95% CI]
Reference/full	T_{80}^{cov}	137.0 [129.0, 145.0]
Reference/without options	T_{80}^{cov}	189.0 [183.4, 194.6]
Reference/without VoI	T_{80}^{cov}	148.0 [143.3, 152.7]
Reference/full	T_{80}^{det}	176.0 [166.2, 185.8]
Reference/without options	T_{80}^{det}	232.0 [224.1, 239.9]
Reference/without VoI	T_{80}^{det}	191.0 [184.3, 197.7]
$N = 8$/full	T_{80}^{cov}	89.0 [85.1, 92.9]
$N = 8$/DTH-MADDPG	T_{80}^{cov}	121.0 [115.1, 126.9]
Constrained/full	Recall	0.900 [0.884, 0.916]
Constrained/TarMAC-style	Recall	0.862 [0.846, 0.878]

The five-seed means preserve the option and VoI contributions and the coverage time advantage at eight UAVs. In the constrained scenario, the complete controller attains 0.900 recall compared with 0.862 for the TarMAC-style comparator, linking receiver-aware evidence selection to cooperative search with obstacles, heterogeneous sensing and mobility, and changing flight restrictions.

5.15. Case study

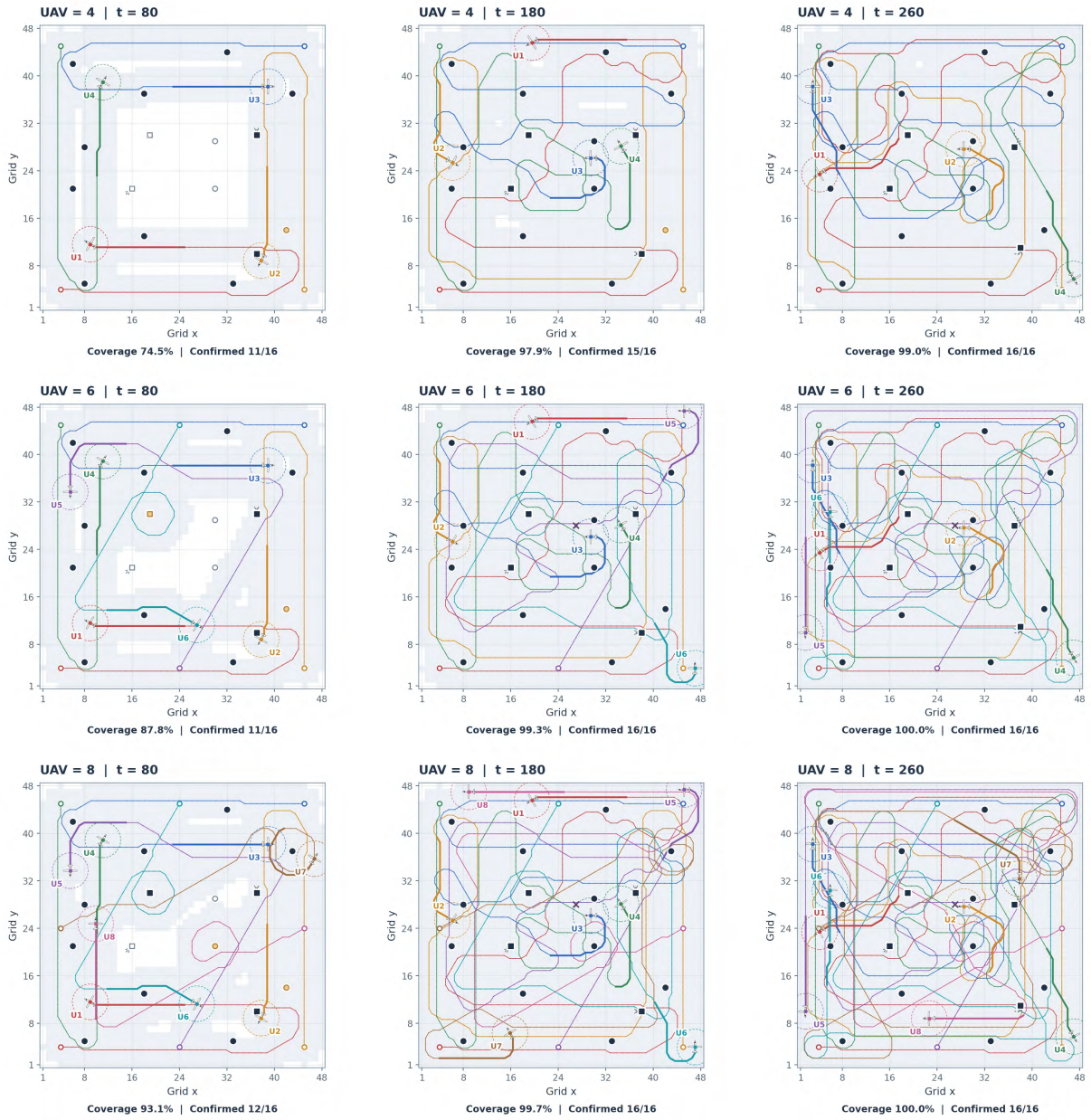


Figure 8. Spatiotemporal evolution of cooperative search for fleets of $N = 4$, 6, and 8 UAVs on the same 48×48 target realization. Rows correspond to the three fleet sizes, while columns show the cumulative trajectories at $t = 80$, 180, and 260. The colored trajectory lines denote UAV-1, UAV-2, UAV-3, UAV-4, UAV-5, UAV-6, UAV-7, and UAV-8.

Figure 8 illustrates how the fleet changes from broad exploration to localized completion during a representative held-out episode. All UAV trajectories are overlaid on the same search area, and every motion segment follows the unit speed action constraint with heading changes restricted to $\{-30°, 0°, +30°\}$. Colored open circles mark the initial positions, aircraft symbols indicate the current positions and headings, and colored dashed circles represent the current sensing footprints. Circular and

square markers denote static and moving targets, respectively. Dark filled markers indicate confirmed targets, amber markers indicate encountered but unconfirmed targets, and hollow markers indicate targets that have not yet been encountered. Light gray shading represents the cumulative area sensed by the fleet, whereas white regions remain unobserved. The values below each panel report the cumulative coverage ratio and the number of confirmed targets. Target locations are shown only as evaluator-side ground truth overlays and are unavailable to the decentralized UAV policies.

At $t = 80$, the UAVs primarily traverse long, separated routes along the boundary and across distinct portions of the interior. As the mission progresses, their paths increasingly enter the remaining unsensed gaps and revisit areas containing unresolved target evidence. This behavior produces broad spatial dispersion during the early phase while still allowing several UAVs to investigate the same high-value region at different times.

For $N = 4$, the fleet covers 74.5% of the grid and confirms 11 of the 16 targets by $t = 80$. Coverage increases to 97.9% at $t = 180$, when 15 targets have been confirmed, and reaches 99.0% with all 16 targets confirmed by $t = 260$. The trajectories show that most of the large-scale sweep is completed during the first two phases. The later motion is concentrated around small interior gaps and previously visited target neighborhoods, reflecting the need to accumulate additional evidence under missed detections, false alarms, and target motion rather than merely maximizing geometric coverage.

Increasing the fleet size mainly improves early coverage and reduces the time required to confirm the final targets. At $t = 80$, coverage rises from 74.5% for four UAVs to 87.8% for six UAVs and 93.1% for eight UAVs; the corresponding numbers of confirmed targets are 11, 11, and 12. By $t = 180$, the six- and eight-UAV fleets have confirmed all 16 targets with coverage rates of 99.3% and 99.7%, whereas the four-UAV fleet still has one unresolved target. At the final timestamp, the six- and eight-UAV fleets achieve complete coverage, while all three fleet sizes confirm every target. The denser path overlap observed for the larger fleets near the end of the mission also suggests diminishing returns once most cells have been sensed and all targets have been localized.

6. Conclusion

We present HOCCA, a hierarchical MARL framework that integrates uncertainty-aware belief estimation, selective communication, and temporally extended control for cooperative multi-UAV searching. Provenance-tagged innovations prevent duplicate information fusion, while receiver-aware VoI scheduling allocates the limited communication budget, and learned options support efficient decentralized execution. Experiments show that HOCCA reduces T_{80}^{cov} and T_{80}^{det} by 25.1% and 23.8%, respectively, relative to DTH-MADDPG, while maintaining efficient coordination across the tested target densities, map scales, and fleet sizes. Future work will examine hardware-in-the-loop sensing and higher-fidelity flight dynamics.

Conflict of interest

The authors declare no competing interests.

Data availability

The simulation environment and evaluation configurations will be made publicly available at https://github.com/wkx12138-w/HOCCASim. The source code of the learning algorithm is available from the corresponding author upon reasonable request.

Author contributions

Jiancheng Liu led the conceptualization and methodology and drafted the manuscript. Wanlong Qi contributed to software implementation and experimental investigation. Yihua Wang contributed to validation and visualization. Bo Li contributed to formal analysis and interpretation of the results. Kexin Wang contributed to methodology and manuscript revision. Dingrui Xue supervised the study and contributed to manuscript revision.

Use of generative AI tools declaration

Generative AI tools were used to assist with programming during the experiments.

Acknowledgments

None.

References

1. J. Wu, J. Luo, C. Jiang, L. Gao, A multiagent deep reinforcement learning approach for multi-UAV cooperative search in multilayered aerial computing networks, *IEEE Internet Things J.*, **12** (2025), 5807–5821. https://doi.org/10.1109/JIOT.2024.3490761
2. J. Liu, S. Wei, B. Li, T. Wang, W. Qi, X. Han, Dual-timescale hierarchical MADDPG for multi-UAV cooperative search, *J. King Saud Univ. Comput. Inf. Sci.*, **37** (2025), 142. https://doi.org/10.1007/s44443-025-00156-6
3. B. O. Koopman, The theory of search. II. Target detection, *Oper. Res.*, **4** (1956), 503–531. https://doi.org/10.1287/opre.4.5.503
4. L. D. Stone, *Theory of optimal search*, Academic Press, 1975. https://shop.elsevier.com/books/theory-of-optimal-search/stone/978-0-12-672450-9
5. H. Yuan, Q. Wang, L. Cui, Z. Xu, Q. Li, S. Yan, Hierarchical reinforcement learning with cohesion-aware adaptive planning for large-scale multi-UAV coverage, *Inf. Sci.*, **753** (2026), 123660. https://doi.org/10.1016/j.ins.2026.123660
6. R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, *Adv. Neural Inf. Process. Syst.*, **30** (2017), 6379–6390. Available from: https://proceedings.neurips.cc/paper-files/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

7. C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, The surprising effectiveness of PPO in cooperative multi-agent games, *Adv. Neural Inf. Process. Syst.*, **35** (2022), 24611–24624. https://doi.org/10.52202/068431-1787
8. T. Rashid, M. Samvelyan, C. S. D. Witt, G. Farquhar, J. Foerster, S. Whiteson, QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning, *Proc. Int. Conf. Mach. Learn.*, 2018, 4295–4304. Available from: https://proceedings.mlr.press/v80/rashid18a.html.
9. X. X. Yang, W. Q. Yao, Y. Zhang, H. Yu, C. Wang, Multi-UAV cooperative search in partially observable low-altitude environments based on deep reinforcement learning, *Drones*, **9** (2025), 825. https://doi.org/10.3390/drones9120825
10. C. C. Ekechi, T. Elfouly, A. Alouani, T. Khattab, A survey on UAV control with multi-agent reinforcement learning, *Drones*, **9** (2025), 484. https://doi.org/10.3390/drones9070484
11. X. Chi, Belief-driven decentralized reinforcement learning for cooperative UAV search in partially observable urban environments, *Int. Acad. Exch. Conf. Sci. Technol. Innov.*, 2025. https://doi.org/10.1109/IAECST68792.2025.11415150
12. X. Zhang, J. Bai, K. Chen, X. Chen, Hierarchical graph-attention multi-agent reinforcement learning for safe-separation-and-collision-avoidance coordination of heterogeneous UAV swarms, *Drones*, **10** (2026), 508. https://doi.org/10.3390/drones10070508
13. Z. Yang, W. Du, X. Zhang, J. Wang, Y. Li, A spatiotemporal graph reasoning approach for pursuit-evasion game with communication limits, *IEEE Trans. Veh. Technol.*, **75** (2026), 2069–2085. https://doi.org/10.1109/TVT.2025.3605736
14. B. Zhang, T. Wang, M. Li, Y. Cui, X. Lin, Z. Zhu, Multiple unmanned aerial vehicle (multi-UAV) reconnaissance and search with limited communication range using semantic episodic memory in reinforcement learning, *Drones*, **8** (2024), 393. https://doi.org/10.3390/drones8080393
15. Y. Liu, X. Li, X. Jiang, H. Liu, J. Yang, Multi-UAV trajectory planning for dynamic target search: An LLM-enhanced multi-agent reinforcement learning algorithm, *IEEE Trans. Cogn. Commun. Netw.*, **12** (2026), 10294–10310. https://doi.org/10.1109/TCCN.2026.3710519
16. J. P. Carvalho, A. P. Aguiar, Multi-agent reinforcement learning for zero-shot coverage path planning with dynamic UAV networks, *Robot. Auton. Syst.*, **195** (2026), 105163. https://doi.org/10.1016/j.robot.2025.105163
17. R. S. Sutton, D. Precup, S. Singh, Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, *Artif. Intell.*, **112** (1999), 181–211. https://doi.org/10.1016/S0004-3702(99)00052-1
18. P. L. Bacon, J. Harb, D. Precup, The option-critic architecture, *Proc. AAAI Conf. Artif. Intell.*, 2017, 1726–1734. https://doi.org/10.1609/aaai.v31i1.10916
19. P. M. Dames, Distributed multi-target search and tracking using the PHD filter, *Auton. Robots*, **44** (2020), 673–689. https://doi.org/10.1007/s10514-019-09840-9
20. F. A. Oliehoek, C. Amato, *A Concise introduction to decentralized POMDPs*, SpringerBriefs in Intelligent Systems, Springer, Cham, Switzerland, 2016. https://doi.org/10.1007/978-3-319-28929-8
21. A. U. Rehman, X. Shi, F. Ullah, Z. Wang, C. Ma, Measuring student attention based on EEG brain signals using deep reinforcement learning, *Expert Syst. Appl.*, **269** (2025), 126426. https://doi.org/10.1016/j.eswa.2025.126426

22. A. U. Rehman, N. U. Eman, F. Ullah, D. Cacciagrano, L. Mostarda, N. Raza, Advanced EEG signal processing and deep Q-learning for accurate student attention monitoring, *IEEE Access*, **13** (2025), 47260–47270. https://doi.org/10.1109/ACCESS.2024.3523225
23. A. U. Rehman, K. Mahmood, E. AbdulKreem, Proactive threat hunting and anomaly detection in wearable EEG consumer devices using double deep reinforcement learning, *IEEE Trans. Consum. Electron.*, 2026. https://doi.org/10.1109/TCE.2026.3695900
24. T. Ali, U. Draz, I. Razzak, M. Hijji, S. Yasin, M. Ayaz, Vision-driven collaborative path planning for autonomous multi-UAV swarms in maritime and coastal environments using deep reinforcement learning, *Robot. Auton. Syst.*, **204** (2026), 105567. https://doi.org/10.1016/j.robot.2026.105567
25. U. Draz, S. Yasin, T. Ali, M. Hijji, M. Ayaz, E. H. M. Aggoune, Communication-aware multi-agent reinforcement learning for cooperative navigation of multiple USVs in dynamic ocean environments, *Ocean Eng.*, **345** (2026), 123626. https://doi.org/10.1016/j.oceaneng.2025.123626
26. U. Draz, S. Yasin, T. Ali, M. Hijji, M. Ayaz, E. H. M. Aggoune, Energy-efficient and interference-resilient optimization of continuous 3D UAV relays in IoT networks using deep deterministic policy gradient, *IEEE Internet Things J.*, 2026. https://doi.org/10.1109/JIOT.2026.3720275
27. H. Zhu, J. Hui, Z. Guo, Two-level-attention-based continuous trajectory design and computation offloading for multi-UAV cooperative target search, *IEEE Trans. Mob. Comput.*, **25** (2026), 3196–3214. https://doi.org/10.1109/TMC.2025.3614596
28. T. D. Kulkarni, K. Narasimhan, A. Saeedi, J. Tenenbaum, Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation, *Adv. neural inf. process. syst.*, 2016. Available from: https://proceedings.neurips.cc/paper-files/paper/2016/hash/f442d33fa06832082290ad8544a8da27-Abstract.html.
29. A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, FeUdal networks for hierarchical reinforcement learning, *Proc. Int. Conf. Mach. Learn.*, 2017, 3540–3549. https://proceedings.mlr.press/v70/vezhnevets17a.html
30. C. Amato, G. Konidaris, A. Anders, G. Cruz, J. P. How, L. P. Kaelbling, Policy search for multi-robot coordination under uncertainty, *Int. J. Robot. Res.*, **35** (2016), 1760–1778. https://doi.org/10.1177/0278364916679611.
31. S. Sukhbaatar, A. Szlam, R. Fergus, Learning multiagent communication with backpropagation, *Adv. Neural Inf. Process. Syst.*, **29** (2016), 2244–2252. Available from: https://proceedings.neurips.cc/paper/2016/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.
32. A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, TarMAC: Targeted multi-agent communication, *Proc. 36th Int. Conf. Mach. Learn.*, **97** (2019), 1538–1546. https://proceedings.mlr.press/v97/das19a.html
33. D. Kim, S. Moon, D. Hostallero, W. J. Kang, T. Lee, K. Son, Learning to schedule communication in multi-agent reinforcement learning, *Proc. Int. Conf. Learn. Represent.*, 2019. Available from: https://openreview.net/forum?id=HUAnBToP-a.
34. Q. Sun, D. Steckelmacher, Y. Yao, A. Nowé, R. Avalos, Dynamic size message scheduling for multi-agent communication under limited bandwidth, *IEEE Trans. Mob. Comput.*, **23** (2024), 15080–15097. https://doi.org/10.1109/TMC.2024.3452986

35. R. Zhao, D. Zhou, S. K. Jha, N. D. Bastian, A. Shah, HiComm: Hierarchical communication for multi-agent reinforcement learning, *arXiv:2606.29126,* 2026. https://doi.org/10.48550/arXiv.2606.29126

36. J. You, Y. Wu, D. Wen, Y. Zhou, Y. Jiang, Y. Shi, Dynamic communication in multi-agent reinforcement learning via information bottleneck, *Proc. IEEE Glob. Commun. Conf.*, 2024, 4884–4889. https://doi.org/10.1109/GLOBECOM52923.2024.10901017

37. V. Mnih, K. Kavukcuoglu, D. Silver, Human-level control through deep reinforcement learning, *Nature*, **518** (2015), 529–533. https://doi.org/10.1038/nature14236

38. H. V. Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, *Proc. AAAI Conf. Artif. Intell.*, **30** (2016), 2094–2100. https://doi.org/10.1609/aaai.v30i1.10295



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)