Research Square

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

Communication-Constrained Dynamic Task Allocation for Heterogeneous UAVs Using Constraint-Aware Cooperative Multi-Agent Proximal Policy Optimization

Shangqing Li
Information Engineering University
Daojie Yu
yudj2003@163.com
Information Engineering University
Yijie Bai
Information Engineering University
Fan Meng
Information Engineering University
Yujie Zhang
Information Engineering University

Research Article

Keywords:

Posted Date: November 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11783042/v1

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

Additional Declarations: No competing interests reported.

Communication-Constrained Dynamic Task Allocation for Heterogeneous UAVs Using Constraint-Aware Cooperative Multi-Agent Proximal Policy Optimization

Shangqing Li[1], Daojie Yu[1*], Yijie Bai[1], Fan Meng[1], Yujie Zhang[1]

[1*]Information Engineering University, Zhengzhou, 450001, Henan, China.

*Corresponding author(s). E-mail(s): yudj2003@163.com;

Abstract

Dynamic task allocation among heterogeneous unmanned aerial vehicles (UAVs) under communication constraints must satisfy coupled task, energy, and communication feasibility requirements. We propose a constraint-aware cooperative Multi-Agent Proximal Policy Optimization (MAPPO) framework that integrates three modular coordination mechanisms: queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M). These mechanisms coordinate task selection, feasibility adjudication, and execution feedback.

Experiments comprise a full factorial ablation over five training seeds and 18 perturbation conditions. In the main scenario, P+R+M raises task completion from 15.75% for baseline MAPPO to 27.33%, i.e. 11.58 percentage points, although P+R alone reaches 27.92%. Across the 18 perturbation conditions, P+R+M averages 22.53%, exceeding P+R by 1.73 points. Rejection memory cuts communication-related rejections by 21.2% but aids task completion only scenario-dependently. Factorial analysis reveals a significant negative R$\times$M interaction (-6.50 percentage points). Batched scoring and observation reuse further cut online allocation time by 60.6% with identical decisions. These results expose both the benefits and the limits of modular constraint coordination.

Keywords: Heterogeneous UAVs; Dynamic task allocation; Multi-agent reinforcement learning; MAPPO; Communication constraints; Task reservation

Categories (2), (3)

1 Introduction

Heterogeneous unmanned aerial vehicle (UAV) swarms can exploit the complementary capabilities of individual platforms in terms of payload capacity, flight speed, and energy resources to cooperatively accomplish time-critical missions, such as emergency search and area inspection. However, in dynamic environments, new tasks may arrive continuously, while previously assigned tasks may be released again due to UAV failures or interruptions during execution. Consequently, task allocation must consider not only task rewards and capability matching but also multiple coupled constraints, including existing task queues, task deadlines, remaining energy, and communication connectivity. Moreover, local observations cannot fully capture the global environmental state, and task proposals generated by individual UAVs may be rejected because of communication limitations or other feasibility constraints. Even when a task proposal is accepted, successful task completion is not guaranteed. Therefore, coordinating task selection with environment-level feasibility adjudication under limited information remains a significant challenge in dynamic multi-UAV task allocation.

Various approaches have been developed for multi-robot task allocation (MRTA), including auction-based methods, consensus-based coordination schemes, and swarm intelligence optimization algorithms [4, 13, 18, 32]. The Consensus-Based Bundle Algorithm (CBBA) coordinates task allocation conflicts through distributed bidding [4], while its extensions further incorporate task time windows and task priorities [27]. Market-based task allocation methods under communication constraints have also investigated task dependencies and information inconsistency [5]. More recently, Multi-Agent Proximal Policy Optimization (MAPPO), typically implemented within the Centralized Training with Decentralized Execution (CTDE) framework, has provided a learning-based approach to dynamic cooperative decision-making [31]. Nevertheless, candidate tasks selected by learned policies may violate complex physical constraints, and independently generated decisions may lead to competition among multiple agents for the same task. Integrating learning-based task selection with multi-UAV coordination mechanisms while effectively exploiting feedback from environment-level feasibility adjudication therefore remains an important research problem.

Recent studies have further investigated MAPPO-based dynamic coalition formation [3], heterogeneous robot task scheduling with constraint lookahead [6], learning-based communication gating mechanisms [30], task submission timeout mechanisms [26], and task coordination under stale information [35]. These studies have provided valuable methodological foundations for dynamic task allocation and communication-aware coordination. However, when locally generated task proposals and environment-level feasibility adjudication are treated as separate processes, the effective utilization of communication rejection feedback remains insufficiently explored. In particular, repeated communication-related rejections may lead to numerous ineffective task proposals, whereas overly restrictive retry mechanisms may prevent UAVs from exploiting feasible allocation opportunities that emerge as communication conditions change. Therefore, reducing ineffective task proposals without compromising eventual task completion is a central challenge addressed in this study.

To address these challenges, this paper proposes a constraint-aware cooperative MAPPO framework for dynamic task allocation among heterogeneous UAVs under communication constraints. The main contributions are summarized as follows.

- A modular constraint coordination layer is developed. The proposed layer comprises three independently configurable mechanisms: queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M). The coordination layer is designed to be decoupled from the underlying policy network and can be integrated with compatible multi-agent task allocation policies through an action-masking interface. In this study, the framework is instantiated and evaluated using CTDE-MAPPO, while its applicability to other learning algorithms remains to be experimentally verified.
- A systematic factorial analysis of the coordination mechanisms is conducted. Using CTDE-MAPPO as the host policy, the individual main effects and interaction effects of P, R, and M are quantified under a unified simulation protocol. The results reveal a statistically significant negative interaction between R and M (-6.50 percentage points), highlighting the importance of accounting for interactions between task reservation and communication feedback mechanisms rather than assuming that their performance gains are additive.
- The modular framework is evaluated under dynamic environmental perturbations. The three coordination mechanisms can be enabled or disabled according to application requirements, allowing different configurations to be selected for different operating conditions. For example, the P+R configuration can be adopted when additional communication rejection memory is not beneficial. The proposed framework is further evaluated under variations in communication radius and UAV failure conditions to examine the scenario-dependent effectiveness of the coordination mechanisms.

2 Related Work

2.1 Traditional Multi-UAV Task Allocation Methods

Traditional approaches to multi-UAV task allocation mainly include auction-based methods, consensus-based coordination schemes, and swarm intelligence optimization algorithms. Zhang et al. proposed the A-QCDPSO algorithm, which integrates an auction mechanism with discrete particle swarm optimization to address dynamic task allocation under unexpected task arrivals and UAV failures [32]. From a problem formulation perspective, Gerkey and Matarić established a formal taxonomy of multi-robot task allocation problems [13], while Korsah et al. further investigated the effects of task dependencies, temporal extensions, and complex constraints on task allocation [18]. These studies provide important theoretical and methodological foundations for dynamic task coordination. Nevertheless, the performance of different approaches remains dependent on factors such as constraint formulations, information availability, and computational search budgets [25].

Recent studies have also investigated task allocation in increasingly complex UAV operating environments. Fan et al. studied cooperative reconnaissance coverage for

heterogeneous UAV swarms under unknown communication interference [9]. Wu et al. applied PG-MAPPO to dynamic task allocation in maritime rescue scenarios [29], while Fu et al. proposed an attention-driven, role-based multi-agent approach for large-scale and highly dynamic task allocation [12]. Furthermore, Ayvaz et al. formulated multi-UAV mission planning as a constraint satisfaction problem with temporal requirements and incorporated dynamic task domains and forward checking to improve resource utilization in multi-task planning [2]. At the mission level, Anastasiou et al. presented an integrated framework for fast and reliable UAV maritime search missions, combining heuristic and optimisation-based planners and validating the framework in real flight experiments [1]. Coordination-layer architectures have also been pursued: Zhang et al. embedded an improved conflict-based search algorithm in an online coordination layer to handle uncertainty in multi-agent environments [34].

2.2 Multi-Agent Reinforcement Learning-Based Task Allocation

Multi-Agent Reinforcement Learning (MARL) enables agents to learn cooperative decision-making policies through interactions with their environments [14]. Lowe et al. introduced a multi-agent actor–critic approach that provides an important foundation for centralized training with decentralized execution [20]. QMIX enables decentralized execution through monotonic value function factorization [24], while COMA employs a counterfactual baseline to address the multi-agent credit assignment problem [10]. Zhang et al. systematically reviewed the theoretical foundations and limitations of centralized training with decentralized execution (CTDE) [33]. To address invalid actions in discrete action spaces, Huang and Ontañón investigated the theoretical properties and practical effectiveness of action masking in policy gradient algorithms [16].

However, under coupled temporal, energy, and communication constraints, local action filtering alone cannot guarantee that selected tasks satisfy all physical feasibility requirements. Regarding energy-aware task allocation, De Alba et al. employed a semi-Markov decision process to model the relationship between UAV battery states and task decisions, highlighting the importance of energy dynamics and task durations in allocation performance [7]. Furthermore, independently generated decisions by multiple agents may result in conflicts when several agents compete for the same task. Consequently, learning-based task selection needs to be integrated with multi-agent coordination mechanisms, environment-level feasibility adjudication, and task execution feedback to improve coordination and feasibility in dynamic task allocation.

2.3 Task Coordination and Feedback under Communication Constraints

Under communication constraints, incomplete information and dynamically changing environmental conditions further complicate multi-UAV task coordination. Qian and Fu investigated uncertainty-gated dynamic task allocation and coordination under intermittent communication conditions [23]. Related studies have sought to improve cooperative decision-making by addressing information consistency, task state management, and communication availability [19, 22]. Foerster et al. investigated learnable

communication protocols in multi-agent reinforcement learning, demonstrating how information exchange among agents can facilitate cooperative decision-making under partial observability [11]. Leong et al. proposed a scalable task allocation approach that considers communication connectivity in flying ad hoc networks, illustrating the close coupling between communication topology and task allocation decisions [19].

Unlike these studies, the present work focuses on dynamic task allocation in which locally generated task proposals and environment-level feasibility adjudication based on complete constraints are treated as distinct processes. By integrating queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M), the proposed framework establishes a coordinated decision-making process that explicitly accounts for task feasibility, same-round reservation conflicts, and communication-related rejection feedback.

Furthermore, this study conducts a full factorial ablation analysis and frozen-actor evaluations to systematically investigate the effects of the proposed mechanisms on task acceptance, eventual task completion, and online computational overhead. These experiments are designed to quantify both the individual contributions of the coordination mechanisms and their interactions under different operating conditions.

3 Problem Formulation

3.1 Dynamic Task Allocation Problem

This study considers dynamic task allocation among heterogeneous UAVs under limited communication ranges, dynamically arriving tasks, and UAV failures. The system consists of multiple UAVs with heterogeneous task execution capabilities and energy resources. Each UAV executes its assigned tasks sequentially according to its existing task queue. During execution, new tasks may arrive, while previously assigned tasks may be released again because of UAV failures or interruptions during task execution. Therefore, the allocation strategy must dynamically adjust task assignments according to the current UAV states, existing task queues, and pending tasks.

Let t denote a discrete physical time step. The set of tasks currently available for allocation is defined as

$$M^R(t) \;=\; M^{\text{new}}(t) \cup M^{\text{release}}(t) \cup M^{\text{pending}}(t), \tag{1}$$

where $M^{\text{new}}(t)$ denotes the set of newly arriving tasks, $M^{\text{release}}(t)$ represents tasks released after execution interruptions, and $M^{\text{pending}}(t)$ denotes previously generated tasks that remain unassigned. These categories distinguish task origins, while the actual candidate set is deduplicated according to task identifiers. Tasks that have already been accepted and remain under normal execution are retained in their corresponding UAV queues and are considered in subsequent feasibility checks, rather than being reassigned or reoptimized at every decision step.

3.2 UAV and Task Models

The UAV operating area is modeled as a two-dimensional plane $\Omega \subseteq \mathbb{R}^2$. The UAV set is denoted by $U = \{u_1, u_2, \ldots, u_N\}$, where N is the total number of UAVs. The state of UAV u_i at time t is represented as

$$s_i(t) = \big(z_i(t),\, \boldsymbol{p}_i(t),\, E_i(t),\, \boldsymbol{c}_i,\, Q_i(t)\big), \tag{2}$$

where $z_i(t) \in \{0, 1\}$ indicates whether UAV u_i is operational, $\boldsymbol{p}_i(t)$ denotes its current position, $E_i(t)$ represents its remaining energy, $\boldsymbol{c}_i$ is its task execution capability vector, and $Q_i(t)$ denotes its queue of accepted but unfinished tasks. The simulation environment also maintains UAV heading information and the progress of the currently executed task to update the physical execution state.

The execution capabilities of each UAV are characterized by a three-dimensional vector

$$\boldsymbol{c}_i = \big[c_i^1,\, c_i^2,\, c_i^3\big]^{\mathsf{T}}. \tag{3}$$

The task set is denoted by $M(t) = \{m_1, m_2, \ldots, m_{J(t)}\}$, where $J(t)$ represents the total number of tasks generated up to time t. Each task m_j is characterized as

$$m_j = \big(\boldsymbol{q}_j,\, \boldsymbol{r}_j,\, V_j,\, \tau_j,\, R_j,\, D_j\big), \tag{4}$$

where $\boldsymbol{q}_j$ denotes the task location, $\boldsymbol{r}_j$ is the three-dimensional capability requirement vector, V_j represents the task reward, τ_j denotes the required service duration, and R_j and D_j represent the task release time and deadline, respectively.

3.3 Task Allocation Constraints

At the k-th allocation round of physical time step t, let $x_{ij}^{t,k} \in \{0, 1\}$ indicate whether UAV u_i proposes to execute task m_j. Since each task proposal must undergo environment-level feasibility adjudication, a separate variable $y_{ij}^{t,k}$ is introduced to indicate whether the proposal is accepted, subject to

$$x_{ij}^{t,k},\, y_{ij}^{t,k} \in \{0, 1\}, \qquad y_{ij}^{t,k} \;\le\; x_{ij}^{t,k}. \tag{5}$$

The task allocation process must satisfy the following constraints.

Task uniqueness constraint

Each UAV can submit at most one task proposal during an allocation round, and each task can be validly reserved by at most one UAV within the same round. Accordingly, the task uniqueness constraints are expressed as

$$\sum_{j} x_{ij}^{t,k} \le 1, \qquad \sum_{i} y_{ij}^{t,k} \le 1, \tag{6}$$

Capability matching constraint

The execution capability of each UAV must satisfy the corresponding task requirements in all three capability dimensions:

$$c_i^d \;\geq\; r_j^d, \qquad d = 1, 2, 3. \tag{7}$$

Temporal and energy constraints

Let $Q_{i'}$ denote the execution queue obtained by appending a candidate task to the existing queue of UAV u_i, and let $T_{i,l}^{\text{fin}}(Q_{i'})$ represent the estimated completion time of task m_l. The temporal and energy feasibility constraints are formulated as

$$\begin{cases} T_{i,l}^{\text{fin}}(Q_{i'}) \leq D_l, & \forall\, m_l \in Q_{i'}, \\ E_i(t) - \hat{E}_i(Q_{i'}) > E_{\text{safe}}, \end{cases} \tag{8}$$

where $\hat{E}_i(Q_{i'})$ represents the estimated energy consumption required to complete the tasks in the updated queue, including flight and service energy consumption, and E_{safe} denotes the minimum safety energy threshold.

Communication feasibility constraint

At the location of a candidate task, at least one other operational UAV must be located within the communication range. This requirement is expressed as

$$\exists\, h \neq i,\; z_h(t) = 1 :\; \left\| \boldsymbol{q}_j - \hat{\boldsymbol{p}}_h(t) \right\|_2 \leq R_c, \tag{9}$$

where R_c denotes the communication radius and $\hat{\boldsymbol{p}}_h(t)$ represents the estimated position of another UAV used for environment-level feasibility adjudication.

3.4 Optimization Objective

The primary optimization objective is to maximize the total reward obtained from tasks successfully completed during an entire simulation episode. Let $\delta_j(T)$ indicate whether task m_j has been completed by the terminal time T. The optimization objective is formulated as

$$\max_{\pi}\; \mathbb{E}_{\pi}\left[\sum_{m_j \in M(T)} V_j\, \delta_j(T) \right], \tag{10}$$

where π denotes the task allocation policy. The optimization is subject to the capability matching, temporal, energy, communication feasibility, and task uniqueness constraints described above.

To evaluate task allocation performance, two metrics are further defined: the task completion rate (CR) and the reward ratio (RR):

$$\mathrm{CR} = \frac{\sum_j \delta_j(T)}{|M(T)|}, \qquad \mathrm{RR} = \frac{\sum_j V_j\, \delta_j(T)}{\sum_j V_j}. \tag{11}$$

The task completion rate measures the proportion of generated tasks that are successfully completed within an episode, while the reward ratio represents the total reward obtained from completed tasks relative to the total reward available from all generated tasks.

4 Proposed Constraint-Aware Cooperative MAPPO Method

It is important to emphasize that the three mechanisms proposed in this study are not specifically designed for MAPPO. The local feasibility filtering mechanism (P) relies only on UAV-specific states and existing task queues, the sequential reservation mechanism (R) utilizes shared reservation information within the current allocation round, and the communication rejection memory mechanism (M) depends solely on rejection feedback returned by the environment. None of these mechanisms requires access to the underlying policy network parameters, observation encoding architecture, or gradient propagation process. Therefore, the proposed coordination layer is designed to be decoupled from the host policy. CTDE-MAPPO is adopted in this study because it provides an established centralized training and decentralized execution framework, together with a standard clipped policy-gradient implementation, facilitating evaluation against baseline methods under a unified experimental protocol.

4.1 CTDE-MAPPO Cooperative Decision-Making Framework

To address coupled feasibility constraints and communication-related task rejections in dynamic task allocation, this study develops a constraint-aware cooperative MAPPO framework, as illustrated in Fig. 1. The proposed approach follows the centralized training with decentralized execution (CTDE) paradigm. A shared actor selects candidate tasks based on local UAV observations, while a centralized critic uses global state information only during training.

During each allocation round, the system sequentially performs local feasibility filtering (P), policy-based task selection, shared task reservation (R), and environment-level feasibility adjudication. Communication rejection feedback is subsequently used to update the rejection memory (M). The actor's action space consists of the currently available candidate tasks and an Idle action. An action mask is applied to exclude invalid choices:

$$\pi_\theta\big(a_i \mid o_i, \boldsymbol{\mu}_i\big) = \mathrm{MaskedSoftmax}\Big(\ell_\theta\big(o_i^{t,k}\big),\, \boldsymbol{\mu}_i^{t,k}\Big)_{a_i}, \tag{12}$$

where θ denotes the shared actor parameters, ℓ_θ represents the network scores assigned to candidate actions, and $\boldsymbol{\mu}_i^{t,k}$ denotes the corresponding valid-action mask.

To improve the feasibility of task proposals, three complementary mechanisms are incorporated into the actor's decision-making process: queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M).

Specifically, P removes candidate tasks that clearly violate locally identifiable execution constraints. R coordinates task selection among UAVs within the same allocation round through shared reservation states, thereby reducing competition for identical tasks. M utilizes communication-related rejection feedback from the environment to suppress repeated infeasible task proposals. Nevertheless, all submitted proposals remain subject to complete environment-level feasibility adjudication. Only tasks that pass the adjudication process and are accepted are appended to the corresponding UAV execution queues.

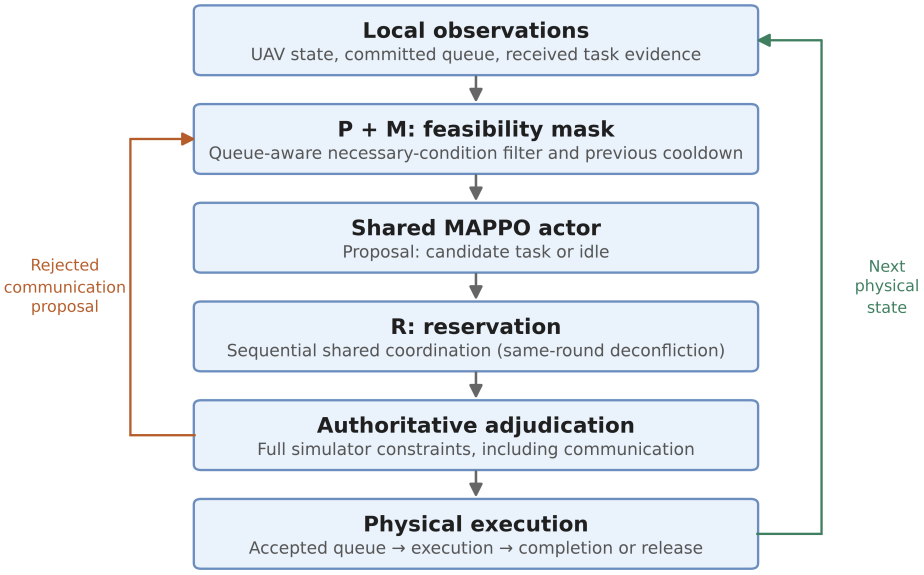


Fig. 1 Constraint-aware proposal, reservation, adjudication, and feedback pipeline

4.2 P: Queue-Aware Local Feasibility Filtering

To reduce task proposals that are clearly infeasible, a local filtering function is constructed using UAV capabilities, existing task queues, temporal constraints, and

remaining energy information:

$$P_{ij}(t) = \begin{cases} 1, & z_i(t) = 1,\ m_j \in M^R(t),\ \boldsymbol{c}_i \geq \boldsymbol{r}_j, \\ & T_{ij}^{\mathrm{fin}}(t) \leq D_j,\ E_{ij}^{\mathrm{after}}(t) > E_{\mathrm{safe}}, \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

where $T_{ij}^{\mathrm{fin}}(t)$ denotes an optimistic estimate of the task completion time after appending the candidate task to the UAV's execution queue, and $E_{ij}^{\mathrm{after}}(t)$ represents the corresponding estimated remaining energy.

This mechanism performs only local necessary-condition checks and does not guarantee complete task feasibility. All remaining constraints, including those requiring additional environmental information, are evaluated through environment-level feasibility adjudication.

4.3 R: Sequential Shared Task Reservation

To prevent multiple UAVs from selecting the same task within a single allocation round, a sequential shared task reservation mechanism is introduced. UAVs select candidate tasks in a predefined order, and tasks already reserved by preceding UAVs are excluded from the candidate sets of subsequent UAVs.

By combining local feasibility filtering, reservation states, and communication rejection memory, the final task-selection mask is defined as

$$\mu_{ij}^{t,k} = P_{ij}(t)\,\big(1 - H_{ij}(t)\big)\,\mathbf{1}\big(j \notin B^{t,k}\big), \tag{14}$$

where $B^{t,k}$ denotes the set of tasks already reserved during the current allocation round, and $H_{ij}(t)$ represents the communication rejection memory state associated with UAV u_i and task m_j.

This mechanism requires a shared reservation service and therefore differs from fully decentralized consensus-based task allocation algorithms [4]. Specifically, the reservation service maintains task reservation information within each allocation round and coordinates task selection among UAVs. Consequently, the proposed framework should not be interpreted as a fully decentralized consensus protocol.

4.4 M: Communication Rejection Memory

When a task proposal passes local feasibility filtering but is rejected during environment-level adjudication because of communication constraints, the system records the rejection time associated with the corresponding UAV–task pair. The same pair is then temporarily excluded from candidate selection through a finite cooldown mechanism:

$$H_{ij}(t) = \begin{cases} 1, & \text{if a valid rejection record exists and } 0 \leq t - \tau_{ij}^{\mathrm{rej}} < \Delta, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where τ_{ij}^{rej} represents the most recent communication rejection time for UAV u_i and task m_j. The cooldown duration Δ is set to six physical time steps in the main configuration.

The cooldown mechanism applies only to UAV–task pairs with recorded communication-related rejections and does not prevent the selection of other feasible tasks. This pair-specific design limits repeated infeasible proposals without globally excluding the corresponding task from all UAVs.

4.5 Policy Training and Online Computational Optimization

The shared actor is trained using the clipped proximal policy optimization (PPO) objective of MAPPO, while a centralized critic estimates the team value function. The actor employs a shared-parameter task-scoring network that separately encodes UAV states and candidate task features, producing selection scores for candidate tasks and the Idle action. The critic estimates the value function using global UAV states, task states, and environmental information.

To balance the final task reward with effective decision-making during task allocation, the following team reward function is employed:

$$r_t = \frac{r_t^{\mathrm{exec}}}{100} + 0.30\, N_t^{\mathrm{acc}} - 0.005\, N_t^{\mathrm{viol}} - 0.003\, N_t^{\mathrm{dup}} - 0.002\, N_t^{\mathrm{idle}} - 0.00005\, E_t^{\mathrm{used}}, \quad (16)$$

where r_t^{exec} denotes the environmental reward obtained during physical task execution, N_t^{acc} is the number of accepted tasks, N_t^{viol} represents the number of invalid allocation decisions, N_t^{dup} denotes the number of duplicate task proposals, N_t^{idle} is the number of times UAVs select the Idle action despite having eligible candidate tasks, and E_t^{used} represents the actual energy consumption.

The reward coefficients are fixed in the main experiments, with weights of 0.30, 0.005, 0.003, 0.002, and 0.00005 for task acceptance, invalid allocation decisions, duplicate task proposals, eligible Idle actions, and energy consumption, respectively. The physical execution reward is normalized by a factor of 100. It should be emphasized that the task acceptance bonus is used for reward shaping during training and does not represent the reward associated with successful task completion.

The optimization objective of the shared actor is expressed as

$$L^{\mathrm{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\big(\rho_t(\theta)A_t,\ \mathrm{clip}\big(\rho_t(\theta), 1-\epsilon, 1+\epsilon\big)A_t\big)\Big], \quad (17)$$

where $\rho_t(\theta)$ denotes the probability ratio between the updated and previous policies under the same conditional action mask, A_t is the advantage estimate computed using generalized advantage estimation (GAE), and ϵ is the PPO clipping parameter.

Policy entropy regularization is incorporated during training, and the critic is updated by minimizing the mean squared error of the value function. For decision states in which Idle is the only available action, the corresponding transitions are excluded from actor policy-gradient updates but retained for critic learning.

During online execution, three computational optimization techniques are introduced to reduce the overhead associated with sequential task selection: forced-Idle skipping, batched candidate task scoring, and observation reuse.

For UAVs without any valid candidate tasks, the Idle action is selected directly, avoiding unnecessary policy network inference. Because the candidate action scores generated by the shared multilayer perceptron (MLP) actor do not depend on the task reservation mask, these scores can be computed in batches. The reservation mask is subsequently updated according to the original sequential decision order, preserving the reservation procedure.

Furthermore, previously computed observations and intermediate results are reused whenever the relevant state information remains unchanged. This reduces redundant feature construction and network inference while preserving the original task-selection decisions.

5 Experimental Setup

5.1 Simulation Environment and Parameter Settings

Experiments are conducted using a simulation platform for dynamic task allocation among heterogeneous UAVs. The simulation environment incorporates heterogeneous UAV capabilities, dynamic task arrivals, accepted task queues, task deadlines, energy consumption, communication constraints, and UAV failures. The main experiments are performed in scenario S20, which consists of eight heterogeneous UAVs operating in a $1000\,\mathrm{m} \times 1000\,\mathrm{m}$ task area, with 10 initial tasks and two dynamically arriving tasks. Each simulation episode lasts 80 physical time steps. UAVs sequentially execute flight and service operations according to their assigned task queues, while unfinished tasks may be released for reassignment following UAV failures or interruptions during execution. The main simulation parameters are summarized in Table 1.

Scenario S20 differs from scenario S2, which is used in the perturbation experiments, primarily in the communication radius. The communication radii are set to 140 m and 100 m for S20 and S2, respectively, while the number of UAVs, the number of tasks, and the capability distributions remain unchanged.

Both UAV execution capabilities and task requirements are represented by three-dimensional vectors. Task deadlines are determined by task release times and randomly generated deadline offsets. Communication feasibility is evaluated based on inter-UAV distances and is continuously checked during task execution. In the main experiments, random communication link failures and message transmission delays are not introduced, allowing the effects of communication range constraints on task allocation performance to be investigated separately.

To evaluate the robustness of the proposed approach under variations in communication range and UAV damage, two scenarios, S20 and S2, are considered. Both scenarios initially contain eight UAVs; therefore, these experiments vary communication conditions and UAV damage levels rather than the initial swarm size. Three communication radius multipliers (0.8, 1.0, and 1.2) and three UAV damage perturbation levels (-1, 0, and $+1$) are considered, resulting in 18 evaluation conditions.

Table 1 Main simulation parameters

Parameter	Value
Task area	1000 m × 1000 m
Number of UAVs	8
Initial / dynamic tasks	10 / 2
Simulation horizon	80 steps
UAV speed	28 m/step
Initial UAV energy	650–950
UAV capability components	0.45–1.00
Task requirement components	0.20–0.85
Task service duration	1–5 steps
Deadline offset	$U(15, 100)$ steps
Flight energy consumption	0.18 units/m
Service energy consumption	8 units/step
Safe energy threshold	40
Communication radius	140 m
Communication stall timeout	5 steps
UAV damage event	1 UAV at step 12
Replanning interval	4 steps
Communication rejection cooldown	6 steps
Actor / Critic learning rate	3×10^{-4} / 8×10^{-4} (linear decay to 25%)
Network architecture and hidden dimensions	Actor: 96; Critic: 128; Idle branch: 48
PPO epochs / mini-batch size	3 / 96
Training hardware	AMD Ryzen Threadripper 3990X (64 cores, 128 threads); NVIDIA TITAN RTX; 128 GB RAM
Total environment interaction steps	80 updates × 256 steps = 20,480 steps per seed

Each of the 18 conditions is evaluated on 10 matched test episodes, giving 180 evaluation episodes per policy. Each of the three learning-based configurations (baseline MAPPO, P+R, and P+R+M) is evaluated with five frozen checkpoints, one per training seed, so each of them aggregates $5 \times 180 = 900$ episodes; each deterministic baseline contributes 180 episodes. In the paired comparisons of Sect. 6, the statistical unit is the training seed: for each seed, the mean over the same 180 matched episodes is compared with the corresponding baseline mean, yielding five paired differences whose interval is a Student t interval with four degrees of freedom.

For these perturbation experiments, the policies trained in scenario S20 are evaluated with the actor network parameters frozen. No additional parameter updates are performed, allowing the performance of the trained policies to be examined under different environmental conditions.

5.2 Baseline Algorithms and Ablation Study

To evaluate the effectiveness of the proposed method, two categories of baseline algorithms are considered.

The first category does not access the environment's complete feasibility information, namely, a feasibility oracle. The representative method in this category is Greedy-Local, a local greedy allocation algorithm that selects the currently executable task with the highest reward based on the complete task list and the current positions and capability information of all UAVs. During its preliminary task selection stage, Greedy-Local does not invoke the environment's complete feasibility checking mechanism.

The second category comprises algorithms that can access global feasibility information and perform feasibility screening before submitting task proposals. This category includes two coordination-based methods, CBBA+JFC [4] and Greedy+JFC, as well as two swarm intelligence optimization algorithms adapted to the current simulation environment, A-QCDPSO [32] and γ-DPSO [28].

In addition, the baseline MAPPO configuration without the three proposed mechanisms ($P = 0$, $R = 0$, $M = 0$) is included as a learning-based reference configuration.

It should be noted that Greedy-Local has access to the complete task list and the state information of all UAVs. Therefore, its information access is not more restricted than that of the proposed actor, making the comparison reported in Table 4 relatively conservative from the perspective of the proposed method.

The complete constraint adjudication mechanism used by the simulator during task execution is referred to as the Joint Feasibility Checker (JFC). Based on the global environmental state, including task queues, remaining energy, task deadlines, capability matching, and communication topology, the JFC determines whether a task proposal can be accepted.

Oracle-assisted baseline algorithms can query the JFC before submitting task proposals and adjust their candidate task sets accordingly. In contrast, the actor in the proposed method relies on local observations and does not access JFC decisions in advance. It obtains feasibility-related information only from previously rejected task proposals and their corresponding rejection reasons.

Johnson et al. emphasized that the design and evaluation of decentralized task allocation methods strongly depend on assumptions regarding information availability, communication conditions, and coordination structures [17]. Therefore, the information access permissions of different algorithms must be explicitly considered when comparing their performance.

Because the two categories of methods differ substantially in information availability, combining them into a single ranking could confound the effects of allocation policy quality and information access. Accordingly, the comparison results are reported separately in Table 4 for methods without access to the feasibility oracle and Table 5 for oracle-assisted reference methods.

All methods are evaluated using the same set of test scenarios and the same environment-level feasibility adjudication rules to ensure comparability of the reported results.

To investigate the individual contributions and interactions of queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M), a full factorial ablation study is conducted with eight possible

configurations. The complete proposed method corresponds to the P+R+M configuration, with the communication rejection cooldown duration fixed at six physical time steps.

5.3 Training Configuration and Evaluation Metrics

All MAPPO ablation configurations employ the same network architecture and training budget. Five independent random seeds are used for training, with 80 parameter updates performed for each seed and 256 environment interaction steps collected per update.

The proximal policy optimization (PPO) clipping parameter is set to 0.2, the discount factor is set to 0.99, and the generalized advantage estimation (GAE) parameter is set to 0.95.

The best-performing model is selected using an independent validation set and subsequently evaluated on a separate held-out test set. Each training seed is evaluated over 20 test episodes.

The primary evaluation metrics include task completion rate, normalized reward ratio, communication rejection count, and episode wall-clock time.

Episode wall-clock time includes task allocation, environment-level feasibility adjudication, and physical simulation. It serves as the common timing metric for comparisons among different allocation algorithms. The independent online computational efficiency experiments are reported separately to avoid mixing results obtained under different timing protocols.

Matched test scenarios are used for paired comparisons, and 95% confidence intervals (CIs) are calculated to assess the performance and stability of the proposed method. For the main learning-based experiments, statistical analyses are performed using independent training seeds [8, 21]. It should be noted that two different types of 95% confidence intervals are reported in this study.

The error bars in Fig. 3(a) represent the variability in task completion rates across 20 matched evaluation scenarios. For the learning-based methods, results obtained from different training seeds are first averaged within each evaluation scenario, after which scenario-level confidence intervals are estimated.

In contrast, Table 5 reports paired 95% confidence intervals for the differences in task completion rates between the proposed method and the corresponding baseline algorithms. These intervals are used to assess performance differences between methods. Therefore, the two types of confidence intervals describe different statistical quantities and should not be used interchangeably.

Henderson et al. highlighted that deep reinforcement learning experiments can be sensitive to random seeds, training stochasticity, and evaluation protocols. Consequently, multiple independent experiments, consistent evaluation conditions, and appropriate statistical analyses are necessary to improve the reliability of experimental conclusions [15].

Additional experiments are conducted to evaluate reservation service reliability and reward weight sensitivity.

For the reservation reliability evaluation, four service interruption probabilities (0%, 5%, 10%, and 20%) are considered. Five trained policies are evaluated under

each interruption condition, with 20 matched test episodes per policy. Communication overhead is estimated using a modeled payload of 24 bytes per reservation request and 16 bytes per acknowledgment or notification message.

For the reward sensitivity analysis, three reward coefficients—the task acceptance bonus, invalid proposal penalty, and eligible Idle penalty—are independently scaled by factors of 0.5 and 2.0, while all remaining coefficients are held fixed.

Each modified reward configuration is trained using three independent random seeds, with 80 parameter updates and 256 environment interaction steps per update. Each trained policy is subsequently evaluated over 20 test episodes.

6 Results and Analysis

6.1 P/R/M Ablation Study

To evaluate the individual contributions and interaction effects of queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M), a full factorial ablation study involving eight configurations is conducted in scenario S20. Each configuration is trained using five independent random seeds, with 20 held-out test episodes evaluated per seed. The task completion rates and normalized reward ratios of the different configurations are presented in Table 2 and visualized in Fig. 2.

Table 2 Performance of P/R/M factorial ablation

Method (P-R-M)	Completion (%)		Reward (%)	
	Mean±SD	95% CI	Mean±SD	95% CI
0-0-0	15.75±2.82	[12.25, 19.25]	17.42±2.43	[14.40, 20.43]
0-0-1	23.42±3.17	[19.49, 27.35]	25.15±3.27	[21.08, 29.21]
0-1-0	25.50±1.94	[23.09, 27.91]	27.36±2.20	[24.62, 30.09]
0-1-1	27.33±1.05	[26.03, 28.63]	29.43±1.38	[27.71, 31.15]
1-0-0	21.00±2.96	[17.33, 24.67]	22.93±2.35	[20.02, 25.84]
1-0-1	27.58±1.02	[26.30, 28.87]	30.10±1.09	[28.74, 31.45]
1-1-0	27.92±1.67	[25.85, 29.99]	29.92±2.03	[27.41, 32.44]
1-1-1	27.33±1.68	[25.24, 29.42]	29.31±2.06	[26.75, 31.86]

Values are mean ± standard deviation over five independent training seeds, with 20 held-out test episodes per seed (within-seed average first, then across seeds). Confidence intervals follow the t distribution with four degrees of freedom. 0/1 indicates whether the corresponding mechanism is enabled.

The complete P+R+M configuration improves the task completion rate from 15.75% for baseline MAPPO to 27.33%, corresponding to an absolute improvement of 11.58 percentage points. The paired 95% confidence interval (CI), calculated across five independent training seeds, is [6.34, 16.83] percentage points. This result indicates that the proposed constraint coordination mechanisms significantly improve task allocation performance compared with baseline MAPPO.

Among the evaluated configurations, P+R achieves the highest task completion rate of 27.92%, whereas P+M obtains the highest normalized reward ratio of 30.10%. These results suggest that the proposed mechanisms interact with one another and that their performance improvements are not necessarily additive.

A further comparison between P+R and P+R+M reveals that incorporating communication rejection memory reduces the average number of communication-related rejections from 69.71 to 54.95 per episode, representing a reduction of approximately 21.2%. However, this reduction does not lead to an additional improvement in task completion rate. This finding indicates that communication rejection feedback can effectively suppress repeated infeasible task proposals, although its contribution to eventual task completion depends on the operating conditions.

Table 3 Marginal effects of main and interaction terms on task completion rate

Effect	Marginal effect (pp)	SD	95% CI
P main effect	+2.96	1.78	[0.75, 5.17]
R main effect	+5.08	1.83	[2.81, 7.35]
M main effect	+3.88	1.38	[2.16, 5.59]
P×R interaction	−3.50	2.96	[−7.17, 0.17]
P×M interaction	−1.75	2.03	[−4.28, 0.78]
R×M interaction	−6.50	2.61	[−9.74, −3.26]
P×R×M interaction	−1.33	6.70	[−9.65, 6.99]

Effects are standard contrasts of the 2^3 full factorial design. SD is the sample standard deviation of the seed-level marginal effect across five training seeds; confidence intervals follow the t distribution with four degrees of freedom.

Table 3 reports the estimated average main effects and interaction effects obtained from the three-factor full factorial design. The average main effects of P, R, and M are +2.96, +5.08, and +3.88 percentage points, respectively. Their corresponding 95% confidence intervals all exclude zero, indicating positive marginal effects when averaged over the levels of the other factors.

However, substantial interactions are observed among the proposed mechanisms. In particular, the interaction between R and M is significantly negative, with an estimated effect of −6.50 percentage points (95% CI: [−9.74, −3.26]). This indicates that the effectiveness of communication rejection memory depends on the presence of the shared task reservation mechanism.

Specifically, when both P and R are enabled, additionally introducing M reduces the task completion rate from 27.92% to 27.33%. Therefore, M does not provide an additional task completion benefit in the main evaluation scenario.

One possible explanation is that shared task reservation already reduces competition for identical tasks within an allocation round, while the fixed cooldown mechanism may further restrict valid retries after communication conditions have recovered. Consequently, some feasible task allocation opportunities may be missed. However, this explanation requires further investigation of retry opportunities suppressed by the memory mechanism.

Overall, these findings demonstrate that the benefits of the proposed mechanisms cannot be assumed to be simply additive and that communication feedback mechanisms must be appropriately coordinated with task reservation strategies.

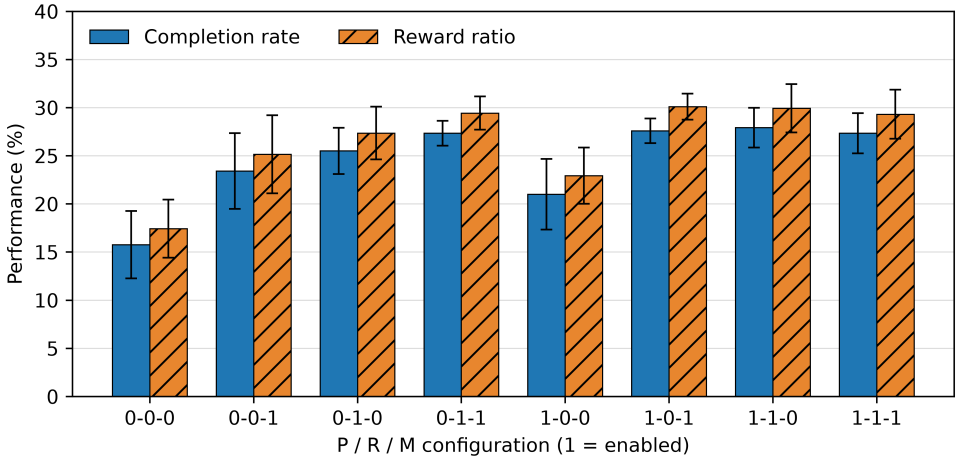


Fig. 2 Performance of the P/R/M factorial ablation on S20. Error bars indicate 95% confidence intervals

6.2 Comparison with Baseline Algorithms

The comparison with baseline algorithms is divided into two categories according to their access to environment-level feasibility information. Table 4 presents the comparison with the non-oracle baseline, whereas Table 5 reports the results of the oracle-assisted reference methods.

Table 4 Performance comparison with the non-oracle baseline

Method	Completion (%)	Episode wall-clock time (s/episode)
Greedy-Local	17.92	0.038
Proposed (P+R+M)	27.33	0.763

Wall-clock time includes policy decisions, task proposals, environment-level feasibility adjudication, and physical simulation.

As shown in Table 4, Greedy-Local achieves a task completion rate of 17.92%, whereas the complete proposed method (P+R+M) achieves 27.33%, corresponding to an improvement of 9.41 percentage points.

The primary difference between these approaches lies in their task selection and feasibility screening procedures. Greedy-Local directly selects the currently executable

Table 5 Comparison with oracle-assisted reference methods

Method	Completion (%)	Wall time (s/ep.)	Proposed − Baseline (pp), 95% CI
CBBA+JFC	28.33	0.123	−1.00 [−3.09, +1.09]
Greedy+JFC	28.75	0.137	−1.42 [−3.51, +0.67]
A-QCDPSO	27.92	112.37	−0.58 [−2.67, +1.51]
γ-DPSO	28.75	160.88	−1.42 [−3.51, +0.67]

All baselines in this table may query the Joint Feasibility Checker before submitting proposals, whereas the proposed actor uses local observations only. The last column reports training-seed-level paired differences.

task with the highest reward based on the global task list, without performing queue-level feasibility filtering before proposal submission. Consequently, many of its task proposals are rejected because of communication constraints.

In contrast, the proposed method integrates local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M) to suppress clearly infeasible or repeated proposals during task selection. As a result, it achieves a higher task completion rate than Greedy-Local without requiring advance access to the environment-level Joint Feasibility Checker (JFC).

Table 5 compares the proposed method with four oracle-assisted task allocation algorithms. CBBA+JFC, Greedy+JFC, A-QCDPSO, and γ-DPSO achieve task completion rates of 28.33%, 28.75%, 27.92%, and 28.75%, respectively, all slightly higher than the 27.33% achieved by the proposed method.

The paired 95% confidence intervals for the task completion rate differences between the proposed method and these four baselines are [−3.09, 1.09], [−3.51, 0.67], [−2.67, 1.51], and [−3.51, 0.67] percentage points, respectively. All four intervals include zero, indicating that no statistically significant differences are detected under the current evaluation protocol and sample size.

Nevertheless, the absence of statistically significant differences should not be interpreted as evidence of statistical equivalence. These statistical conclusions are based on paired differences in task completion rates rather than the overlap of the individual confidence intervals displayed in Fig. 3.

Furthermore, the oracle-assisted algorithms can query the complete environment-level JFC before submitting task proposals, whereas the proposed actor relies on local observations and rejection feedback without accessing JFC decisions in advance. However, the proposed sequential reservation mechanism still requires a shared coordination service.

Therefore, these comparisons primarily illustrate the trade-offs among information availability, computational budgets, and task allocation performance rather than demonstrating that the proposed method outperforms all traditional algorithms across every metric.

Regarding computational overhead, episode wall-clock time is adopted as the common timing metric to ensure consistency across different algorithms. This metric includes policy decisions, task proposals, environment-level feasibility adjudication, and physical simulation, rather than measuring policy network inference alone.

Under the same evaluation protocol, the proposed method requires an average of 0.763 s per episode, approximately 6.2 times and 5.6 times the wall-clock times of CBBA+JFC (0.123 s) and Greedy+JFC (0.137 s), respectively. By comparison, A-QCDPSO and γ-DPSO require 112.37 s and 160.88 s per episode, approximately 147 and 211 times the wall-clock time of the proposed method.

These results demonstrate that the proposed method offers a substantial computational efficiency advantage over iterative swarm intelligence optimization algorithms, although its computational cost remains higher than that of lightweight coordination-based methods.

A further analysis of task proposal behavior reveals that baseline MAPPO generates an average of 164.03 task proposals per episode, of which 2.61 are accepted. After incorporating the P, R, and M mechanisms, the average number of proposals decreases to 60.70, representing a reduction of approximately 63.0%, while the average number of accepted proposals increases to 5.75, corresponding to an increase of approximately 120.3%.

These findings indicate that the proposed constraint coordination mechanisms not only reduce ineffective task proposals but also increase the number of successfully accepted tasks.

Nevertheless, the proposed method still generates approximately 60.70 task proposals per episode, whereas the four traditional baseline algorithms generate only 4.70–5.10 proposals, all of which are accepted.

Because each proposal requires environment-level checks of task queues, temporal feasibility, energy constraints, and communication feasibility, the relatively high proposal frequency may be an important reason why the proposed method incurs greater episode-level computational overhead than lightweight coordination algorithms.

Therefore, the main computational advantage of the proposed method lies in its efficiency relative to iterative search-based optimization algorithms, together with its ability to suppress ineffective proposals compared with baseline MAPPO, rather than achieving the lowest computational cost among all evaluated methods.

It should also be noted that the evaluated algorithms differ in information access, decision-making procedures, and computational budgets. Oracle-assisted methods can access complete environment information and employ batch-oriented optimization procedures, whereas the proposed method performs sequential online decisions for individual UAVs.

Accordingly, this study does not claim that the proposed method is superior across all performance metrics. Instead, it demonstrates improved task completion relative to the non-oracle greedy baseline and competitive performance compared with oracle-assisted approaches, despite not accessing the feasibility oracle in advance.

6.3 Analysis under Communication Radius and UAV Failure Perturbations

To evaluate the adaptability of the proposed approach to environmental variations, frozen-actor comparisons are conducted under 18 evaluation conditions formed by two scenario types, three communication radius multipliers, and three UAV failure

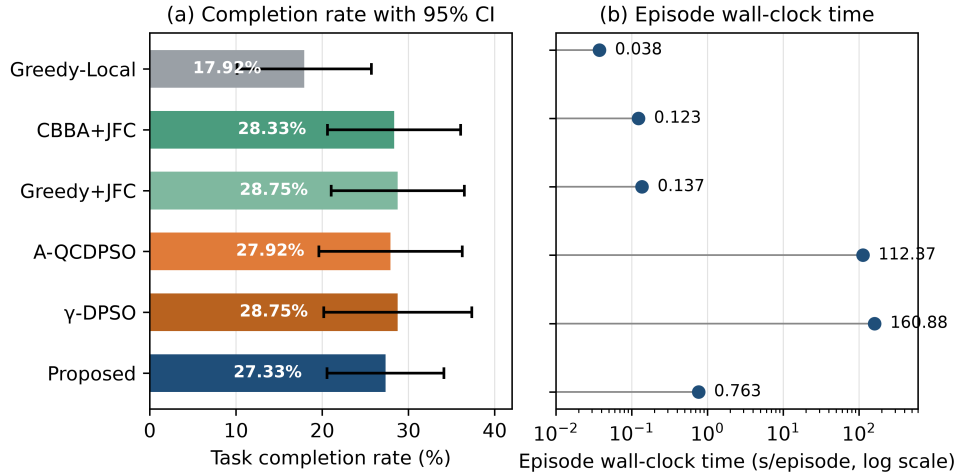


Fig. 3 Comparison of (a) task completion rates with scenario-level 95% confidence intervals and (b) episode wall-clock times on a logarithmic scale

perturbation levels; the resulting completion rates are shown in Fig. 4. The actor network parameters remain unchanged throughout these evaluations.

The results show that baseline MAPPO, P+R, and the complete P+R+M method achieve average task completion rates of 13.32%, 20.80%, and 22.53%, respectively.

Compared with baseline MAPPO, P+R+M improves the task completion rate by 9.20 percentage points. Compared with P+R, the improvement is 1.73 percentage points, with a paired 95% confidence interval of [0.73, 2.73] percentage points.

The proposed method is also compared with traditional baseline algorithms under the same 18 perturbation conditions. The average task completion rates are 23.56% for γ-DPSO, 23.38% for A-QCDPSO, 22.53% for P+R+M, 22.08% for Greedy+JFC, and 21.62% for CBBA+JFC. Each deterministic baseline is evaluated over 180 matched episodes. For the proposed method, five frozen checkpoints (one per training seed) are each evaluated on the same 180 episodes, yielding 900 checkpoint–episode evaluations in total; the paired intervals below use the training seed as the statistical unit.

Based on comparisons paired by training seed, the proposed method achieves a significantly higher average task completion rate than CBBA+JFC (+0.91 percentage points, 95% CI: [0.28, 1.53]). No statistically significant difference is detected relative to Greedy+JFC (+0.44 percentage points, 95% CI: [−0.18, 1.07]).

However, the proposed method performs significantly worse than both swarm intelligence baselines: compared with A-QCDPSO the difference is −0.85 percentage points (95% CI: [−1.47, −0.23]), and compared with γ-DPSO it is −1.04 percentage points (95% CI: [−1.66, −0.41]).

These results indicate that, when communication ranges and UAV failure levels vary, the relative competitiveness of the proposed method lies between lightweight coordination-based approaches and iterative swarm intelligence optimization algorithms.

The results further suggest that communication rejection memory can provide additional performance benefits under certain environmental perturbations. Nevertheless, when considered together with the S20 ablation results, the effect of this mechanism on task completion is clearly scenario-dependent.

Therefore, a reduction in communication-related rejection counts does not necessarily translate into a corresponding improvement in eventual task completion.

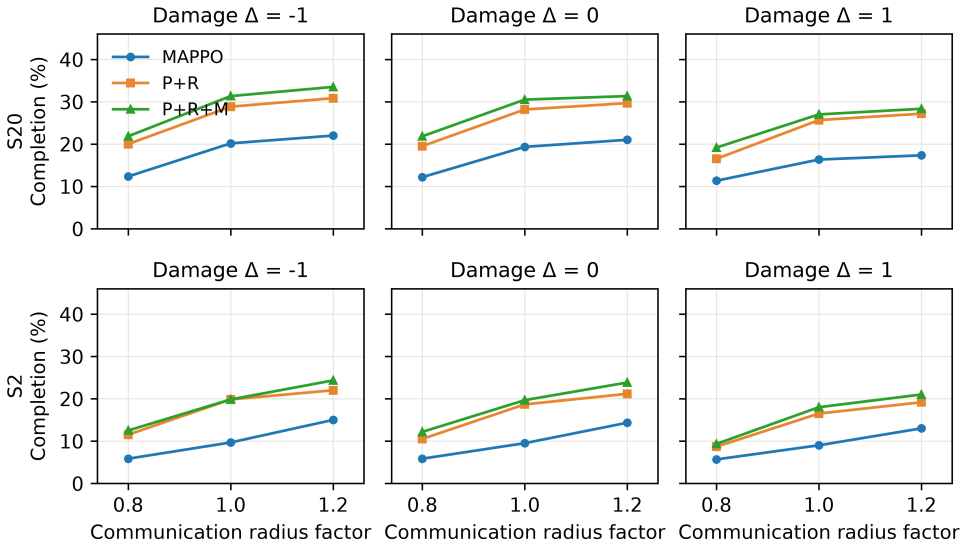


Fig. 4 Task completion rates of frozen policies across 18 communication and UAV-damage conditions

6.4 Online Computational Efficiency Analysis

To reduce the computational overhead of the shared actor network during sequential decision-making, two optimization techniques are introduced: batched candidate task scoring and observation reuse. An independent paired timing experiment is conducted using frozen actor networks to evaluate their effectiveness.

In this experiment, the online allocation computation times before and after optimization are 1769.15 ms and 697.68 ms, respectively, representing a reduction of approximately 60.6%. Meanwhile, task proposals, action selections, and final task execution outcomes remain unchanged.

It should be emphasized that this experiment employs a separate evaluation scenario set and timing protocol. Its measured quantity is the computational cost of the online allocation procedure, which differs from the episode wall-clock time reported in Tables 4 and 5.

Consequently, the optimized computation time of 697.68 ms should not be directly compared with or substituted for the 0.763 s episode wall-clock time reported in the baseline comparison.

These findings demonstrate that batched candidate scoring and observation reuse can reduce software-level computational overhead without changing policy decisions. However, the observed reduction does not imply an equivalent improvement in actual wireless communication overhead or end-to-end task execution time.

6.5 Reservation Service Outages and Communication Overhead

Because the sequential shared task reservation mechanism (R) relies on a shared coordination service, its communication overhead and potential failure risks must be evaluated.

Five trained policies are tested under four reservation service interruption probabilities: 0%, 5%, 10%, and 20%. Each policy is evaluated over 20 matched test episodes under each interruption condition.

Service interruptions are applied at the allocation-round level. When the shared reservation service is unavailable, the reservation restriction for the affected allocation round is disabled, following a fail-open strategy, while environment-level feasibility adjudication remains active.

The modeled communication overhead assumes a 24-byte reservation request and a 16-byte acknowledgment or notification message. Table 6 summarizes the results.

Table 6 Reservation service outages and communication overhead

Interruption prob.	Completion (%)	95% CI	Repeated props.	Payload (B/ep.)
0%	30.92	[29.58, 32.26]	0.00	7407
5%	30.92	[29.35, 32.49]	2.90	7114
10%	30.83	[29.08, 32.59]	5.39	6907
20%	30.17	[28.35, 31.98]	11.08	6571

Under ideal coordination conditions, the estimated coordination payload is approximately 7.41 kB per episode, and no duplicate task proposals are observed.

When the interruption probability increases to 20%, the average number of duplicate proposals rises to 11.08 per episode, while the task completion rate decreases from 30.92% to 30.17%. The corresponding completion rate difference is -0.75 percentage points, with a paired 95% confidence interval of $[-1.75, 0.25]$ percentage points.

Because the confidence interval includes zero, no statistically significant reduction in task completion rate is detected within the tested range and sample size.

These results indicate that reservation service outages increase task proposal conflicts, although their effect on eventual task completion remains limited under the simulated conditions.

Importantly, the reported coordination payload represents only an estimate of modeled coordination messages. It does not include protocol headers, physical-layer retransmissions, broadcast overhead, or existing task announcements, and therefore should not be interpreted as the actual number of bytes transmitted over a wireless channel.

Moreover, a logical reservation service interruption is not equivalent to real wireless packet loss.

6.6 Reward Weight Sensitivity Analysis

The manually selected coefficients in the team reward function may influence policy performance. To investigate this effect, sensitivity experiments are conducted on three reward coefficients: the task acceptance bonus, invalid proposal penalty, and eligible Idle penalty. The results are shown in Fig. 5 and Table 7.

Each coefficient is independently scaled by factors of 0.5 and 2.0, while all other reward coefficients remain unchanged. Each modified configuration is trained using three independent random seeds and evaluated over 20 test episodes per seed.

This sensitivity analysis uses an independent evaluation set and a different reference policy group from those used in the main ablation study. Therefore, the absolute completion rates reported here should not be directly compared with the results in Table 2.

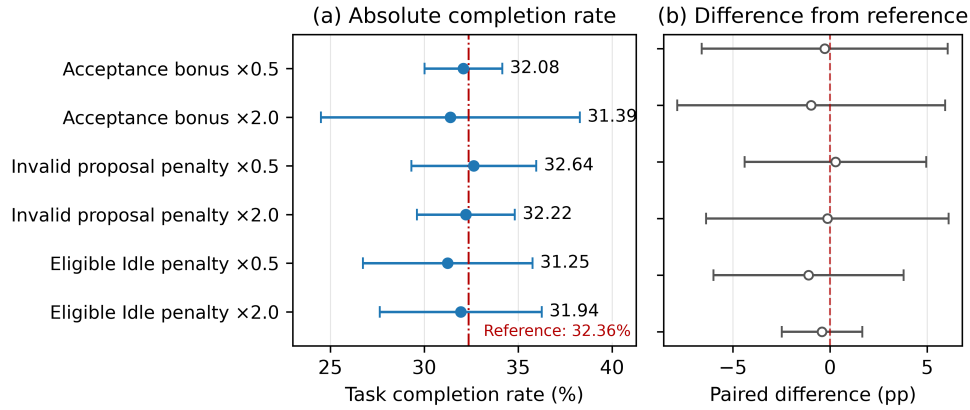


Fig. 5 Sensitivity of task completion rate to reward-weight variations

Table 7 Reward weight sensitivity analysis

Reward configuration	Completion (%)	Paired diff. vs. reference, 95% CI (pp)
Reference (1×)	32.36	—
Acceptance bonus (0.5×)	32.08	[−6.60, +6.05]
Acceptance bonus (2.0×)	31.39	[−7.86, +5.92]
Invalid proposal penalty (0.5×)	32.64	[−4.39, +4.95]
Invalid proposal penalty (2.0×)	32.22	[−6.38, +6.10]
Eligible Idle penalty (0.5×)	31.25	[−6.00, +3.78]
Eligible Idle penalty (2.0×)	31.94	[−2.49, +1.65]

The reference reward configuration achieves a task completion rate of 32.36%, while the six modified configurations yield completion rates ranging from 31.25% to 32.64%.

The paired 95% confidence intervals for the completion rate differences between all modified configurations and the reference configuration include zero. This indicates that no statistically significant effect of the tested reward weight variations on task completion rate is detected under the current evaluation protocol and sample size.

However, the sensitivity analysis covers only three reward coefficients and uses three independent training seeds, resulting in relatively wide confidence intervals. Therefore, these findings do not establish that the proposed method is insensitive to all reward coefficients, nor do they demonstrate statistical equivalence among different reward configurations.

The energy consumption penalty is not included in the present sensitivity experiments. Furthermore, because no duplicate task proposals are observed during normal P+R+M training, the duplicate proposal penalty is not independently varied.

7 Discussion and Limitations

From a methodological perspective, the primary contribution of this study is not a new multi-agent reinforcement learning algorithm, but rather a modular constraint coordination layer and its system-level evaluation for dynamic task allocation among heterogeneous UAVs under communication constraints. The proposed framework integrates queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M) to coordinate task proposals, environment-level feasibility adjudication, and the utilization of rejection feedback. The ablation results show that the complete P+R+M configuration improves the task completion rate by 11.58 percentage points over baseline MAPPO. However, its completion rate in the main scenario is slightly lower than that of P+R, indicating that the benefits of the individual mechanisms are not simply additive.

Further analysis reveals that communication rejection memory reduces communication-related rejections by approximately 21.2% but does not provide an additional improvement in task completion rate in the main scenario. One possible explanation is that the fixed cooldown mechanism, while suppressing repeated infeasible proposals, may also prevent valid retries after communication conditions recover, thereby causing missed task allocation opportunities. The three-factor interaction analysis identifies a significant negative interaction between R and M, with an estimated effect of -6.50 percentage points (95% CI: $[-9.74, -3.26]$). This suggests that the combined effectiveness of shared reservation and rejection memory depends on their interaction. Nevertheless, across the 18 communication-radius and UAV-failure perturbation conditions, P+R+M achieves an average task completion rate 1.73 percentage points higher than P+R. Taken together, these findings indicate that the practical benefit of communication rejection memory is scenario-dependent. Future research should therefore investigate adaptive cooldown and retry strategies that respond to changes in communication and task coordination conditions, rather than relying exclusively on a fixed cooldown duration.

Despite the evidence provided by the ablation, cross-scenario, and supplementary experiments, the proposed approach has several limitations. First, communication feasibility is modeled primarily using geometric proximity. Specifically, a candidate task satisfies the communication feasibility condition when at least one other operational UAV is located within the communication radius of the candidate task position. The main experiments do not account for stochastic packet loss, message transmission delays, asymmetric communication links, or multihop relaying. Although logical interruptions of the shared reservation service and the resulting coordination message payload have been evaluated, these experiments are not equivalent to actual wireless link failures and do not cover physical failures of the coordination node. Therefore, the reported results primarily characterize task allocation performance under geometric communication constraints and modeled reservation service interruptions, rather than establishing robustness under realistic wireless communication conditions.

Second, the evaluation scale remains limited. The main experiments involve eight heterogeneous UAVs, ten initial tasks, and two dynamically arriving tasks. Although this setting captures task competition, queue coupling, and communication-related rejection feedback, it does not fully represent the allocation conflicts, communication topology variations, and coordination service loads that may arise in swarms consisting of tens of UAVs. Consequently, the present results should not be extrapolated to larger UAV swarms without further experimental validation.

Moreover, the proposed approach still incurs considerable task proposal and environment-level adjudication overhead. Although P, R, and M reduce the average number of task proposals from 164.03 per episode for baseline MAPPO to 60.70, the proposal count remains substantially higher than the 4.70–5.10 proposals observed for the traditional coordination-based and optimization-based baselines. Because each proposal requires complete environment-level feasibility checks, the relatively high proposal frequency increases computational overhead and may partly explain why the proposed method has a higher episode wall-clock time than lightweight coordination algorithms. Future work should investigate more effective preliminary feasibility filtering, explicit proposal budgets, and reuse of environment-level adjudication results to reduce redundant computations and improve practical deployability on resource-constrained UAV platforms.

Finally, all experiments are conducted in simulation, without deployment tests on physical UAV platforms or real wireless networks. The episode wall-clock times reported in the baseline comparisons include policy decisions, environment-level feasibility adjudication, and physical simulation, whereas the online allocation computation times reported in the independent efficiency experiments are measured using a different timing protocol. Both measurements depend on the specific software implementation and hardware platform and should not be interpreted as end-to-end communication or decision latency in real UAV systems. Future work will include stochastic link failures and communication delays, scalability experiments with different swarm sizes, and hardware-in-the-loop or real-world validation under realistic communication conditions to more comprehensively assess the applicability, robustness, and deployment potential of the proposed framework.

8 Conclusions

This paper proposes a constraint-aware cooperative Multi-Agent Proximal Policy Optimization (MAPPO) approach to address coupled feasibility constraints and ineffective task proposals in dynamic task allocation among heterogeneous unmanned aerial vehicles (UAVs) under communication constraints. The proposed framework integrates queue-aware local feasibility filtering (P), sequential shared task reservation (R), and communication rejection memory (M) to coordinate task selection, environment-level feasibility adjudication, and execution feedback.

Experimental results demonstrate that the complete P+R+M configuration improves the task completion rate from 15.75% for baseline MAPPO to 27.33%, corresponding to an increase of 11.58 percentage points. Across 18 perturbation conditions, P+R+M achieves an average task completion rate 1.73 percentage points higher than that of P+R. In addition, batched candidate scoring and observation reuse reduce online allocation computation time by approximately 60.6% while preserving identical decision outcomes.

The findings indicate that constraint coordination and communication rejection feedback can improve the task allocation performance of baseline MAPPO, although the benefits of individual mechanisms depend on the operating conditions. The three-factor analysis further reveals a significant negative interaction between sequential shared reservation and communication rejection memory, suggesting that feedback mechanisms should be designed in conjunction with the coordination process. Future research will investigate adaptive communication feedback strategies and further evaluate the effectiveness and robustness of the proposed framework under more complex communication conditions and in larger heterogeneous UAV swarms.

References

[1] Anastasiou A, Papaioannou S, Kolios P, et al (2026) A framework for fast and reliable UAV maritime search missions. Journal of Intelligent & Robotic Systems https://doi.org/10.1007/s10846-026-02448-z, in press

[2] Ayvaz E, Atay Y, Babaoğlu İ (2025) A cutting-edge approach to multi-UAV mission planning using enhanced constraint satisfaction. Journal of Intelligent & Robotic Systems 111:95. https://doi.org/10.1007/s10846-025-02279-4

[3] Bezerra LCD, dos Santos AMG, Park S (2025) Learning policies for dynamic coalition formation in multi-robot task allocation. IEEE Robotics and Automation Letters 10(9):9216–9223. https://doi.org/10.1109/LRA.2025.3592080

[4] Choi HL, Brunet L, How JP (2009) Consensus-based decentralized auctions for robust task allocation. IEEE Transactions on Robotics 25(4):912–926. https://doi.org/10.1109/TRO.2009.2022423

[5] Chu J, Yan F, Xu Z (2025) Communication-constrained multi-UAV task allocation method for non-independent tasks. Engineering Science and Technology, an

International Journal 70:102166. https://doi.org/10.1016/j.jestch.2025.102166

[6] Dai W, Rai U, Chiun J, et al (2025) Heterogeneous multi-robot task allocation and scheduling via reinforcement learning. IEEE Robotics and Automation Letters 10(3):2654–2661. https://doi.org/10.1109/LRA.2025.3534682

[7] De Alba A, Flores A, García Maya B, et al (2025) Optimizing UAV task allocation with enhanced battery efficiency using semi-Markov decision processes. Journal of Intelligent & Robotic Systems 111:86. https://doi.org/10.1007/s10846-025-02285-6

[8] Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7:1–30

[9] Fan Y, Chen B, Zhao Y, et al (2025) Cooperative reconnaissance coverage for heterogeneous unmanned aerial vehicle swarm in unknown communication interference environments. Engineering Applications of Artificial Intelligence 159:111717. https://doi.org/10.1016/j.engappai.2025.111717

[10] Foerster J, Farquhar G, Afouras T, et al (2018) Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 2974–2982, https://doi.org/10.1609/aaai.v32i1.11794

[11] Foerster JN, Assael YM, de Freitas N, et al (2016) Learning to communicate with deep multi-agent reinforcement learning. In: Advances in Neural Information Processing Systems, pp 2137–2145

[12] Fu H, Wang J, He L, et al (2026) Task assignment method of heterogeneous UAV swarm based on role multi-agent reinforcement learning. Aeronautical Science & Technology 37(2):1–7. https://doi.org/10.19452/j.issn1007-5453.2026.02.001, (in Chinese)

[13] Gerkey BP, Matarić MJ (2004) A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research 23(9):939–954. https://doi.org/10.1177/0278364904045564

[14] Hady MA, Hu S, Pratama M, et al (2025) Multi-agent reinforcement learning for resources allocation optimization: a survey. Artificial Intelligence Review 58:354. https://doi.org/10.1007/s10462-025-11340-5

[15] Henderson P, Islam R, Bachman P, et al (2018) Deep reinforcement learning that matters. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 3207–3214, https://doi.org/10.1609/aaai.v32i1.11694

[16] Huang S, Ontañón S (2022) A closer look at invalid action masking in policy gradient algorithms. In: Proceedings of the International FLAIRS Conference, https://doi.org/10.32473/flairs.v35i.130584

[17] Johnson LB, Choi HL, How JP (2016) The role of information assumptions in decentralized task allocation: a tutorial. IEEE Control Systems Magazine 36(4):45–58. https://doi.org/10.1109/MCS.2016.2558419

[18] Korsah GA, Stentz A, Dias MB (2013) A comprehensive taxonomy for multi-robot task allocation. The International Journal of Robotics Research 32(12):1495–1512. https://doi.org/10.1177/0278364913496484

[19] Leong WL, Cao J, Teo R (2024) Scalable task allocation with communications connectivity for flying ad-hoc networks. Journal of Intelligent & Robotic Systems 110:25. https://doi.org/10.1007/s10846-024-02059-6

[20] Lowe R, Wu Y, Tamar A, et al (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, pp 6379–6390

[21] Nadeau C, Bengio Y (2003) Inference for the generalization error. Machine Learning 52:239–281. https://doi.org/10.1023/A:1024068626366

[22] Oliehoek FA, Amato C (2016) A Concise Introduction to Decentralized POMDPs. Springer International Publishing, Cham, https://doi.org/10.1007/978-3-319-28929-8

[23] Qian C, Fu B (2026) Uncertainty-gated distributed dynamic task allocation for multi-UAV systems under intermittent communication. Drones 10(10):782. https://doi.org/10.3390/drones10100782

[24] Rashid T, Samvelyan M, Schroeder de Witt C, et al (2018) QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: Proceedings of the 35th International Conference on Machine Learning, PMLR 80, pp 4295–4304

[25] Simukonda L, Santos L, Santos FN, et al (2026) Multi-robot path planning in agriculture: benchmarking state-of-the-art MAPF algorithms. Journal of Intelligent & Robotic Systems https://doi.org/10.1007/s10846-026-02402-z, in press

[26] Sun P, Yang G, Xu X, et al (2026) Distributed task allocation algorithm for heterogeneous UAVs based on reinforcement learning. Drones 10(3):220. https://doi.org/10.3390/drones10030220

[27] Wang K, Wang D, Li M, et al (2026) E-CBBA: a consensus auction approach for multi-UAV task allocation under time window and task priority constraints. IEEE Transactions on Automation Science and Engineering 23:14713–14724. https://doi.org/10.1109/TASE.2026.3722233

[28] Wu Q, Guo J, Kang Z, et al (2024) Maritime mission assignment of UAV clusters based on γ random search strategy. Journal of Beijing University of

Aeronautics and Astronautics 50(12):3872–3883. https://doi.org/10.13700/j.bh.1001-5965.2022.0882, (in Chinese)

[29] Wu X, Yan Q, Wang J, et al (2025) Dynamic task allocation for UAV swarms in maritime rescue scenarios based on PG-MAPPO. IEEE Internet of Things Journal 12(18):38073–38087. https://doi.org/10.1109/JIOT.2025.3584767

[30] Xiong Z, Xi Y, Cao Y, et al (2026) Task allocation with communication coordination in UAV swarms via asynchronous multi-objective policy optimization. Aerospace Science and Technology 174:111760. https://doi.org/10.1016/j.ast.2026.111760

[31] Yu C, Velu A, Vinitsky E, et al (2022) The surprising effectiveness of PPO in cooperative multi-agent games. In: Advances in Neural Information Processing Systems (NeurIPS 35), pp 24611–24624, https://doi.org/10.52202/068431-1787

[32] Zhang J, Chen Y, Yang Q, et al (2022) Dynamic task allocation of multiple UAVs based on improved A-QCDPSO. Electronics 11(7):1028. https://doi.org/10.3390/electronics11071028

[33] Zhang K, Yang Z, Başar T (2021) Multi-agent reinforcement learning: a selective overview of theories and algorithms. In: Vamvoudakis KG, Wan Y, Lewis FL, et al (eds) Handbook of Reinforcement Learning and Control. Springer, p 321–384, https://doi.org/10.1007/978-3-030-60990-0_12

[34] Zhang S, Zeng Q, Zeng Y (2026) An online coordination layer based on improved conflict based search algorithm in uncertain environments. Journal of Intelligent & Robotic Systems 112:105. https://doi.org/10.1007/s10846-026-02416-7

[35] Zhao C, Li M, Zhu X, et al (2026) Assignment-consistent dynamic multi-UAV task allocation: communication-efficient dynamic multi-unmanned aerial vehicle task allocation under stale and asymmetric information. Drones 10(7):523. https://doi.org/10.3390/drones10070523

Statements and Declarations

Funding

The authors declare that no specific funding was received for this study.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Ethics approval

Not applicable. This study did not involve human participants or animals.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Data availability

The simulation configuration files and the per-episode result files that support the figures and tables of this study are available from the corresponding author on reasonable request.

Code availability

The simulation environment, the training scripts and the baseline implementations used in this study are available from the corresponding author on reasonable request.

Author contributions

S. Li: conceptualization, methodology, software, formal analysis, investigation, data curation, visualization, writing – original draft. **D. Yu**: supervision, methodology, writing – review and editing. **Y. Bai**: software, data curation, validation, visualization. **F. Meng**: investigation, resources, writing – review and editing. **Y. Zhang**: validation, formal analysis, writing – review and editing. All authors read and approved the final manuscript.