machines

MDPI

Article

Resilient Multi-UAV Mission Execution Under Partial Capability Degradation via Event-Triggered Task Reallocation

Qingyun Yang [1,2], Yewei Zhang [1,*], Kun Yang [1], Zhenyang Xu [3] and Shuyi Shao [3]

[1] School of Mechanical and Electrical Engineering, Zaozhuang University, Zaozhuang 277160, China; yqysmee@uzz.edu.cn (Q.Y.)
[2] Jinzhong Hydraulic Rubber Hose Co., Ltd., Zaozhuang 277000, China
[3] College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
* Correspondence: zhangyewei2015@163.com

Abstract

A UAV may remain able to service targets after its motion-execution capability declines, yet its current assignment can become inefficient. We propose Event-Triggered Capability-Aware Task Reallocation (ETCR), a centralized task layer coupled to a standard multi-agent proximal policy optimization (MAPPO) motion policy. ETCR uses a speed-based travel-cost proxy and switching and workload penalties to reassess current assignments at mission events. In planar missions with four UAVs and six targets, the original five-pair confirmation yielded a severe-degradation success gain of 5.10 percentage points (pp; bootstrap 95% interval 2.80–6.60). Five additional training pairs yielded +7.70 pp on the original cases; all ten pairs on a new fixed case bank yielded +9.05 pp (7.20–10.80). With identical frozen actors, replacing distance allocation with ETCR improved severe success by 4.3 and 4.1 pp for MAPPO-origin and ETCR-origin actors, respectively. A separate contrast holding penalties and triggers fixed attributed 6.2 and 5.9 pp to the speed-aware cost. Isolated degradation tests support speed-related adaptation; stable response-only and additional soft-trigger benefits were not established. The results demonstrate the value of updating task guidance from speed-related capability information within the tested hierarchical execution setting.

Keywords: multi-UAV systems; multi-robot task allocation; partial capability degradation; resilient mission execution; event-triggered reallocation; MAPPO

1. Introduction

Sequential missions involving multiple unmanned aerial vehicles (UAVs) require a team to decide which vehicle should pursue each outstanding objective and how each vehicle should move toward its current objective. Multi-robot task allocation (MRTA) formalizes the first decision and distinguishes, among other dimensions, single-task from multi-task robots and instantaneous from time-extended assignment problems [1,2]. A recent multi-UAV survey identifies task arrivals, vehicle failures, and changing mission requirements as distinct sources of allocation dynamics [3]. A disruption during execution, however, need not be a binary transition from healthy operation to complete failure. A UAV may remain active and able to service the same targets while a reduced command-speed limit or slower dynamic response lowers its execution efficiency. Under this partial degradation, the original task structure can remain feasible but cease to be an efficient organization of the team.

Check for updates

Academic Editor: Carmine Maria Pappalardo

Received: 9 October 2026
Revised: 27 October 2026
Accepted: 30 October 2026
Published: 10 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Existing MRTA research already provides substantial foundations for heterogeneous, dynamic, and resilient coordination. Market-based mechanisms support online coordination across changing task sets [4], while consensus-based auctions provide decentralized conflict resolution and robustness to inconsistent situational awareness and communication topology changes [5]. Recent approaches include distributed scheduling with cross-schedule dependencies [6] and behavior-tree capability models with runtime auctions [7]. For changing assignments, reoptimization heuristics reuse existing solutions after task insertion or deletion [8], while DTA-HMR-TT transfers tasks among heterogeneous mobile robots to handle delivery timing constraints [9].

Resilient allocation methods have used observed task-performance deviations to revise capability-dependent resource assignments under adverse conditions [10]; energy-aware online frameworks have represented heterogeneous robot features and capabilities to retain mission survivability under environmental impairment and component failures [11]. Recent formulations also explicitly address task- or time-dependent variable capabilities under temporal-logic and inter-task constraints [12], and long-endurance heterogeneous missions can repair or recompute plans online when unexpected events occur [13]. In multi-UAV missions, Wang et al. combine reliability-constrained pre-assignment with contract-net reassignment after execution disruptions [14]. Jiang et al. incorporate health information into a learned communication mechanism for exploration with degradation, failure, and rescue [15]. These works motivate adaptation to changing capability, while addressing broader scheduling or recovery settings than the task-feasible efficiency loss considered here.

Recent studies also address residual capability and assignment continuity. Liu et al. use local-information auction groups and damage-state representations for resilient swarm reallocation [16]; this shares the need to reorganize a damaged but still useful team, while emphasizing distributed information and recovery. Zeng et al. study payload-failure resilience through bidirectional task reallocation [17]. Within this established line of research, ETCR focuses on scalar motion-execution efficiency, with target-service feasibility retained and speed-related information entering current assignment costs.

STAP^2 addresses decentralized spatiotemporal task allocation and path planning for time-sensitive multi-UAV missions [18]. ETCR instead studies a centralized current-target matching coupled to a fixed MAPPO motion interface, without claiming a complete spatiotemporal route schedule. Assignment-Preserving Replanning inherits the execution state at an event, protects valid assignments and relative route order, and selectively releases protected tasks when feasibility repair is insufficient [19]. ETCR shares the concern for assignment continuity, but its switch penalty acts on a current matching rather than protecting the structure of a residual route plan. The comparison concerns problem formulation and execution interfaces; the present experiments do not rank these planners.

Event-triggering has also been applied directly to task coordination: graph reinforcement learning has been coupled with event-triggered synchronization for dynamic multi-UAV assignment under weak communication [20], while adaptive consensus has initiated task negotiation after significant events and has been compared with periodic coordination [21]. Hu et al. switch between joint allocation–path optimization and path planning for urban moving-target tracking using an event-triggered mechanism [22]. Against this background, we examine how speed-related capability information can update current task guidance when an affected UAV remains task-feasible. Event-triggering schedules reassessment; communication reduction is outside the present objective.

This issue is architectural in a hierarchical decision system. The task layer determines *where to go*, whereas the motion layer determines *how to move*. Integrated task-and-motion planners explicitly couple discrete allocation with continuous feasibility [23], and loosely coupled fleet architectures combine online assignment, motion planning, coordination, and

control [24]. D-ITAGS further interleaves allocation, scheduling, and motion planning and selectively repairs the solution components affected by disruptions [25]. This establishes targeted repair across planning layers as an existing approach; ETCR instead reassesses the current target assignment while retaining standard MAPPO motion-policy optimization.

Multi-agent reinforcement learning offers a complementary route to continuous cooperative decisions through centralized training and decentralized execution [26]; multi-agent proximal policy optimization (MAPPO) is a strong on-policy baseline built on proximal policy optimization [27,28]. Learning also supports task-level decisions: Dai et al. learn decentralized task selection and scheduling for heterogeneous coalitions [29], and ATA-HRL uses hierarchical reinforcement learning for initial allocation and conditional reallocation in heterogeneous human–robot teams [30]. Hyper-SAMARL coordinates task allocation and socially aware navigation through a hypergraph-based MARL architecture [31], while RL-TAPU uses distinct learning components for allocation and path-finding under environmental uncertainty [32]. These studies establish adaptive and hierarchical allocation as prior capabilities. Here, the question is whether partial-degradation information should be absorbed by the learned motion actor or used explicitly to restructure the current task assignment.

We address this question with Event-Triggered Capability-Aware Task Reallocation (ETCR). An externally supplied capability change is evaluated at the task layer; a mission-relevant event invokes assignment reassessment; the resulting rectangular linear assignment selectively changes current task guidance only when the updated cost favors a different organization. A trained, standard MAPPO policy then produces continuous velocity commands for the current guidance target. Event-triggered control has traditionally tied controller updates to informative state changes [33]; ETCR applies the scheduling principle to discrete task reassessment rather than modifying the motion controller. The name describes when the allocator is invoked; it does not imply that every trigger or penalty independently improves mission success. The MAPPO actor, critic loss, clipped surrogate objective, and optimization procedure are intentionally left unchanged.

The contributions are as follows:

1. A capability-aware task evaluation maps partial speed degradation to an interpretable speed-based travel-cost proxy, thereby translating an execution-efficiency change into an online task preference without treating the operational UAV as failed.
2. An event-triggered selective reallocation mechanism reassesses a centralized, receding current assignment after initialization, task completion, hard dropout, or a significant soft capability change. Reassessment does not force reassignment, and the task-layer mechanism remains decoupled from standard MAPPO policy optimization.
3. A systematic evaluation combines architectural comparisons, common frozen-policy allocator diagnostics, and independent training-seed replication to assess the contribution of speed-aware task reallocation under partial capability degradation.

2. Problem Formulation

Consider a team of UAVs indexed by

$$\mathcal{R} = \{1, \dots, N\}, \tag{1}$$

and a set of stationary mission targets

$$\mathcal{G} = \{1, \dots, M\}. \tag{2}$$

At decision step t, UAV i has planar position $\boldsymbol{p}_{i,t} \in \mathbb{R}^2$, while target j is located at $\boldsymbol{g}_j \in \mathbb{R}^2$. Let $b_{j,t} \in \{0,1\}$ indicate whether target j has been completed and define the remaining-target set as

$$\mathcal{J}_t = \{j \in \mathcal{G} \mid b_{j,t} = 0\}. \tag{3}$$

Once completed, a target is removed permanently from $\mathcal{J}_t$.

The time-varying capability state of UAV i is

$$\boldsymbol{h}_{i,t} = \begin{bmatrix} k^v_{i,t} & k^r_{i,t} & \chi_{i,t} \end{bmatrix}^\mathsf{T}, \tag{4}$$

where $k^v_{i,t}$ is the command-speed and steady-state capability ratio, $k^r_{i,t}$ is the dynamic-response ratio, and $\chi_{i,t} \in \{0, 1\}$ denotes whether the UAV is active. Nominal operation corresponds to $\boldsymbol{h}_{i,t} = [1, 1, 1]^\mathsf{T}$. Under partial degradation,

$$\chi_{i,t} = 1, \qquad k^v_{i,t} < 1 \text{ and/or } k^r_{i,t} < 1, \tag{5}$$

so the UAV remains task-feasible but becomes efficiency-degraded. Hard dropout is represented by $\chi_{i,t} = 0$. Fault diagnosis is outside the scope of this work: $\boldsymbol{h}_{i,t}$ is assumed to be supplied by an external health or capability-monitoring module.

The capability variables affect execution in distinct ways. Let $\tilde{\boldsymbol{a}}_{i,t} \in \mathbb{R}^2$ be the raw motion-policy output for an active UAV. The frozen implementation constructs the bounded velocity command as

$$s_{i,t} = \min\left(1, \frac{k^v_{i,t}\bar{v}}{\max(\|\tilde{\boldsymbol{a}}_{i,t}\|_2, \epsilon_a)}\right), \tag{6}$$

$$\boldsymbol{a}_{i,t} = s_{i,t}\tilde{\boldsymbol{a}}_{i,t} = \begin{bmatrix} v^{\mathrm{cmd}}_{x,i,t} & v^{\mathrm{cmd}}_{y,i,t} \end{bmatrix}^\mathsf{T}, \qquad \|\boldsymbol{a}_{i,t}\|_2 \le k^v_{i,t}\bar{v}, \tag{7}$$

where $\bar{v}$ is the nominal command-speed limit and $\epsilon_a = 10^{-12}$ is the command-norm floor used by the clipping operation. The simulator adopts a first-order velocity-response approximation, using the standard exponential response and exact zero-order-hold discretization of linear systems ([34], Chapter 5). The response ratio sets the time constant and decay factor as

$$\rho_{i,t} = \max(k^r_{i,t}, \epsilon_h), \qquad \tau_{i,t} = \frac{\bar{\tau}}{\rho_{i,t}}, \qquad \beta_{i,t} = \exp\left(-\frac{\Delta t}{\tau_{i,t}}\right), \tag{8}$$

where $\bar{\tau}$ is the nominal time constant and $\epsilon_h > 0$ is the response-ratio floor. Holding $\boldsymbol{a}_{i,t}$ and the capability state constant over the step, the exact discrete updates of this simplified model for the actual velocity $\boldsymbol{v}_{i,t}$ and position are

$$\boldsymbol{v}_{i,t+1} = \boldsymbol{a}_{i,t} + (\boldsymbol{v}_{i,t} - \boldsymbol{a}_{i,t})\beta_{i,t}, \tag{9}$$

$$\boldsymbol{p}_{i,t+1} = \boldsymbol{p}_{i,t} + \boldsymbol{a}_{i,t}\Delta t + \tau_{i,t}(\boldsymbol{v}_{i,t} - \boldsymbol{a}_{i,t})(1 - \beta_{i,t}). \tag{10}$$

The command bound is not an instantaneous hard bound on actual velocity after a soft capability reduction. The existing velocity state is retained and may temporarily exceed the reduced command limit while relaxing through the response dynamics. Thus, a smaller $k^r_{i,t}$ increases $\tau_{i,t}$ and slows the actual-velocity response without changing the command bound. The frozen experiments use $\Delta t = 0.2$ s, $\bar{v} = 2.0$ workspace units per second, $\bar{\tau} = 0.5$ s, and $\epsilon_h = 0.05$. Inactive UAVs receive the zero command and are excluded from motion and allocation.

The active set is $\mathcal{A}_t = \{i \in \mathcal{R} \mid \chi_{i,t} = 1\}$. A binary variable $x_{ij,t}$ denotes whether target j is the current organizational target of UAV i. The receding assignment satisfies

$$\sum_{j \in \mathcal{J}_t} x_{ij,t} \le 1, \qquad i \in \mathcal{A}_t, \tag{11}$$

$$\sum_{i \in \mathcal{A}_t} x_{ij,t} \le 1, \qquad j \in \mathcal{J}_t, \tag{12}$$

$$\sum_{i \in \mathcal{A}_t} \sum_{j \in \mathcal{J}_t} x_{ij,t} = \min(|\mathcal{A}_t|, |\mathcal{J}_t|), \qquad x_{ij,t} \in \{0, 1\}. \tag{13}$$

These constraints define a one-to-one current matching, consistent with the single-task robot and single-robot task categories in MRTA taxonomies [1,2]. Thus, each UAV holds at most one current target and no future route or task bundle is preplanned. When more targets than active UAVs remain, only a subset is currently assigned, and the other targets stay pending. When fewer targets remain, some active UAVs are unassigned. Such UAVs still receive actor commands with a zero target-relative input; they are not forced to hover and may incidentally service a target. The centralized assignment is recomputed recedingly from the current state.

Figure 1 illustrates the current-assignment and pending-target semantics for the four-UAV, six-target mission used in the experiments. Partial degradation leaves the affected UAV operational and its current target feasible, but changes the relative execution cost and therefore motivates reassessment. The dashed alternative in Figure 1c is a candidate considered during that reassessment, not a simultaneous assignment or a prescribed transfer.

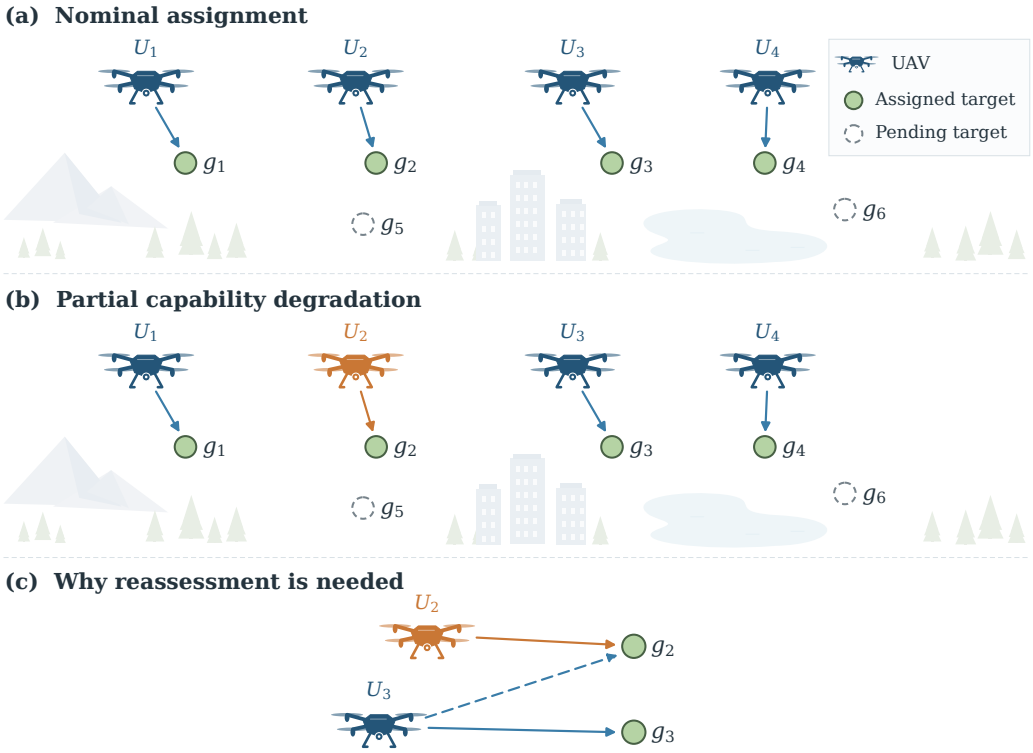


Figure 1. Partial capability degradation in sequential missions involving multiple unmanned aerial vehicles (UAVs). (**a**) Under nominal operation, g_1–g_4 are currently assigned and g_5–g_6 are pending. (**b**) UAV U_2 remains operational and task-feasible after partial degradation while retaining g_2 as its current assignment. (**c**) Solid arrows denote current assignments; the dashed U_3-to-g_2 arrow denotes an alternative candidate considered during reassessment, which need not change the assignment.

The current assignment provides guidance for team organization, not exclusive service authorization. At each motion step, any active UAV that enters the service radius of a remaining target may complete it; if several UAVs are within the radius, the closest is credited. This execution rule permits incidental service by a UAV other than the currently assigned one and is retained unchanged by ETCR.

The resulting hierarchy separates responsibilities. ETCR consumes the monitored capability state and selects the current organizational target—the *where to go* decision. A motion policy trained using standard MAPPO [28] maps local kinematics, current-target relation, mission progress, and neighbor information to the continuous command in (7)—the *how to move* decision. For the ETCR configuration, true capability values are

not supplied to the decentralized actor; the actor retains nominal capability placeholders, while the centralized critic used during training receives the true global state. This interface preserves the standard MAPPO optimization and isolates adaptation at the task layer.

3. Event-Triggered Capability-Aware Task Reallocation

The hierarchical mission-execution architecture is summarized in Figure 2. Capability estimates are assumed to be supplied by an external monitoring module and enter ETCR together with the current mission state. ETCR determines *where to go* by evaluating and, when an event warrants it, reassessing the current task assignment; the assigned target is then passed to the standard MAPPO policy, which determines *how to move* through continuous velocity commands. True capability values are not supplied to the MAPPO actor in the ETCR configuration, and MAPPO optimization remains unchanged.

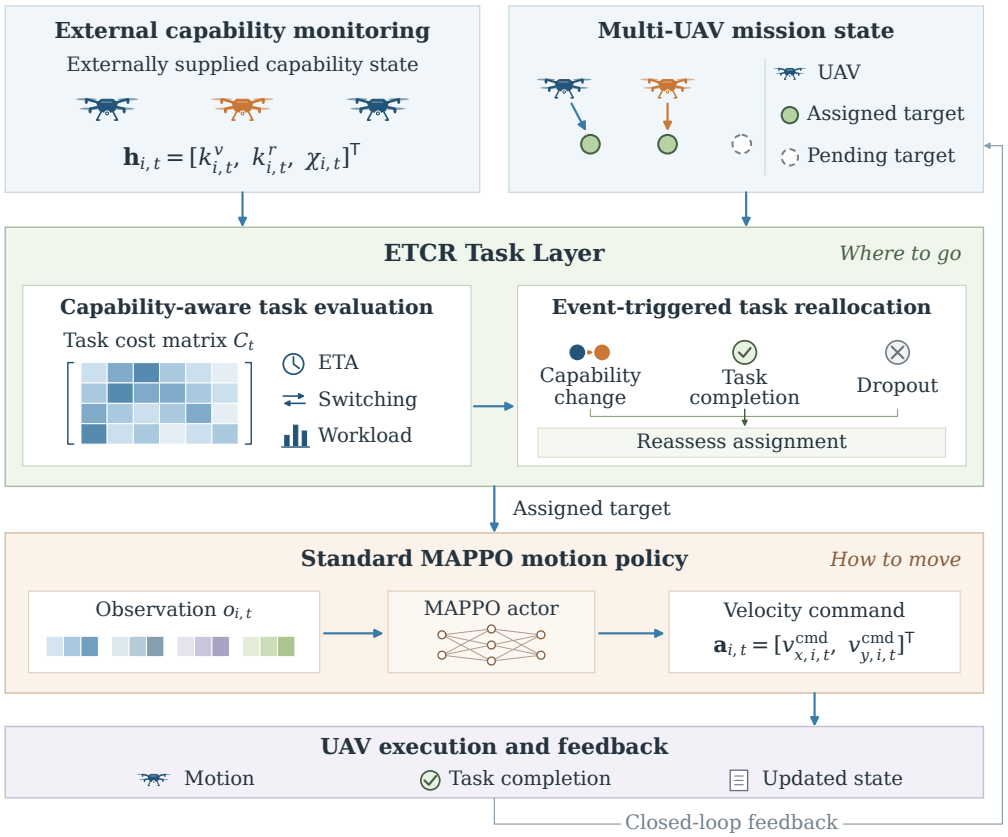


Figure 2. Hierarchical mission-execution architecture with the Event-Triggered Capability-Aware Task Reallocation (ETCR) task layer and standard multi-agent proximal policy optimization (MAPPO) motion policy. External capability estimates and the current mission state support task evaluation and event-triggered assignment reassessment; assigned targets then condition the standard MAPPO actor, whose velocity commands drive UAV execution and close the loop through state and completion feedback. ETA denotes a speed-based travel-cost proxy, not exact arrival time under the response dynamics.

3.1. Capability-Aware Task Evaluation

Let

$$d_{ij,t} = \left\| \boldsymbol{p}_{i,t} - \boldsymbol{g}_j \right\|_2. \tag{14}$$

The distance-only allocator used with standard MAPPO assigns the current targets with

$$C^{\mathrm{dist}}_{ij,t} = d_{ij,t}. \tag{15}$$

Capability-dependent allocation has been studied under degraded or variable capabilities [10,12]. For the present task-feasible setting, ETCR adopts the following specific speed-based travel-cost proxy (retaining the historical ETA notation) for its travel component:

$$\mathrm{ETA}_{ij,t} = \frac{d_{ij,t}}{\max\left(k_{i,t}^{v}\bar{v}, \epsilon_{v}\right)}, \tag{16}$$

where $\epsilon_v > 0$ prevents division by zero. This term is a speed-based travel-cost proxy, not an exact ETA: it omits the current velocity, response time constant, turning, controller actions, and service radius. From rest under a constant saturated one-dimensional command $u = k^v\bar{v}$, the actual center-arrival time instead satisfies $d = u[T - \tau(1 - \exp(-T/\tau))]$. That idealized relation does not itself model the complete two-dimensional closed loop. Equation (16) explicitly uses the command-speed ratio $k_{i,t}^{v}$. The response ratio $k_{i,t}^{r}$ does not enter the ETA or any other allocator-cost term directly: it affects the physical dynamics through (8)–(10), can contribute to a capability-change event, and can influence later allocation states indirectly through position and accumulated path length. The alive indicator $\chi_{i,t}$ only determines the active candidate set.

Dynamic task planning can retain unaffected parts of an existing solution through targeted repair [25]. Here, ETCR encourages assignment continuity through its own switching penalty. Let $q_{i,t}^{-} \in \mathcal{T}_t \cup \{-1\}$ be the target held by UAV i immediately before reassessment. We define the switching indicator as

$$P_{ij,t}^{\mathrm{sw}} = \begin{cases} 1, & q_{i,t}^{-} \in \mathcal{T}_t \text{ and } j \neq q_{i,t}^{-}, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

It regularizes unnecessary assignment changes but does not guarantee stability. The implementation also computes a normalized UAV-level workload

$$w_{i,t} = \frac{1}{2}\left(\frac{n_{i,t}^{\mathrm{done}}}{\max(1, \max_{\ell \in \mathcal{R}} n_{\ell,t}^{\mathrm{done}})} + \frac{L_{i,t}}{\max(1, \max_{\ell \in \mathcal{R}} L_{\ell,t})} \right), \tag{18}$$

where $n_{i,t}^{\mathrm{done}}$ is the completed-target count and $L_{i,t}$ is accumulated path length. The ETCR cost is

$$C_{ij,t} = \mathrm{ETA}_{ij,t} + \lambda_{\mathrm{s}} P_{ij,t}^{\mathrm{sw}} + \lambda_{\mathrm{l}} w_{i,t}. \tag{19}$$

The workload component is constant across the candidate-target row of a given UAV. Consequently, when every active UAV receives a target, this row-wise term need not change the UAV–target permutation. It can affect which UAVs receive targets when fewer targets than active UAVs remain. It is therefore treated as a limited agent-selection regularizer rather than as a claim of continuous global workload balancing.

Remark 1 (Monotonicity)**.** *For $d_{ij,t} > 0$ and the operating region $k_{i,t}^{v}\bar{v} > \epsilon_v$,*

$$\frac{\partial\, \mathrm{ETA}_{ij,t}}{\partial k_{i,t}^{v}} = -\frac{d_{ij,t}}{\left(k_{i,t}^{v}\right)^{2}\bar{v}} < 0. \tag{20}$$

Thus, partial speed degradation monotonically increases the estimated execution cost until the numerical speed floor becomes active.

The switching indicator and normalized workload are dimensionless; the frozen weights are $\lambda_{\mathrm{s}} = 0.5$ s and $\lambda_{\mathrm{l}} = 0.75$ s. The unit floor in path-length normalization represents one workspace unit. When every active UAV receives one target, the workload contribution is a fixed row sum and cannot change the set of minimum-sum matchings. It can affect which UAVs are selected when fewer targets than active UAVs remain; no global load-balancing guarantee is implied.

3.2. Event-Triggered Task Reallocation

Event-triggered controller updates [33] and event-triggered task negotiation [21] provide precedents for condition-dependent updates. ETCR applies this idea to task-assignment reassessment, with the capability-change threshold below defined for this method; it does not trigger motion-controller updates. ETCR invokes the centralized allocator at four mission-relevant events: initialization, completion of one or more targets, hard dropout, and a significant soft capability change. For an active UAV, a soft event is significant when

$$\left\|\boldsymbol{h}_{i,t}^{+} - \boldsymbol{h}_{i,t}^{-}\right\|_2 > \delta_h, \tag{21}$$

where the superscripts denote the capability vectors immediately after and before the scheduled change. Hard dropout is handled explicitly and does not rely on the threshold test. The evaluated scheduler applies at most one scheduled jump per episode. Equation (21) does not guarantee detection of accumulated subthreshold deterioration: six decreases of 0.1 each remain below a threshold of 0.15 despite a cumulative decrease of 0.6. Completion events can still cause reassessment. A response-only change can also invoke reassessment without directly changing the cost at a fixed state; it is not direct response-aware cost optimization.

At a reassessment, ETCR solves

$$\underset{\{x_{ij,t}\}}{\text{minimize}} \quad \sum_{i \in \mathcal{A}_t} \sum_{j \in \widetilde{\mathcal{J}}_t} C_{ij,t} x_{ij,t} \tag{22}$$

subject to (11)–(13). This is a standard rectangular linear assignment problem; algorithms for this problem class are reviewed by Crouse [35]. The sum objective is a receding current-assignment surrogate and does not guarantee minimum complete-mission makespan or finite-horizon success. For example, with the current travel-cost matrix

$$C = \begin{bmatrix} 2 & 6 \\ 6 & 9 \end{bmatrix},$$

the sum-optimal matching has maximum edge cost 9, whereas the alternative has maximum 6. A linear assignment routine computes the matching. The solver is an implementation choice, not a methodological contribution. Because (17) preserves progress information and the new optimum may equal the old one, an event triggers reassessment but not forced reassignment.

Algorithm 1 summarizes online mission execution. Its ordering matches the frozen implementation: the initial assignment precedes any event scheduled at the initial boundary; a due capability event is applied before the next motion command; target service is evaluated after motion; and any completion is followed by assignment reassessment. Applying an event changes the capability state before the associated dropout or soft-change trigger is evaluated.

The frozen allocator parameters are $\epsilon_v = 10^{-6}$, $\lambda_s = 0.5$, $\lambda_1 = 0.75$, and $\delta_h = 0.15$; their roles and the remaining environment parameters are documented in Section 4.1 and Supplementary Section S1. MAPPO training pseudocode is omitted because ETCR retains the existing PPO/MAPPO policy optimization [27,28]. Algorithm 1 specifies the integration of that motion policy with the ETCR cost and selective task-reassessment mechanism.

Algorithm 1 Event-Triggered Capability-Aware Task Reallocation for Online Mission Execution

Input: trained standard MAPPO policy π_θ, initial UAV and target states, event schedule, δ_h, and allocator parameters.
Output: completed-target state and online assignment record.

1. Initialize $b_{j,0} = 0$, $q^-_{i,0} = -1$, and $\boldsymbol{h}_{i,0} = [1,1,1]^\mathsf{T}$; solve (22) with trigger initialization.
2. At the current decision boundary, apply a scheduled capability event if one is due.
3. If the event sets $\chi_{i,t} = 0$, zero the affected UAV's actual velocity and previous command and reassess over the surviving UAVs. Otherwise, if (21) holds, reassess using the updated capability state.
4. Draw the raw output $\tilde{\boldsymbol{a}}_{i,t} \sim \pi_\theta(\cdot \mid o_{i,t})$, obtain the bounded velocity command from (6) and (7), and propagate the actual velocity and position with (9) and (10).
5. Mark every remaining target within service radius as completed by its closest active UAV. If at least one target is completed, reassess all remaining targets with trigger task completion.
6. Advance the decision step. If the mission is not terminated or truncated, return to Step 2; otherwise, return the mission record.

4. Experimental Results

4.1. Experimental Setup

The frozen study instantiates $N = 4$ UAVs and $M = 6$ stationary service targets in the abstract planar workspace $[-10, 10]^2$; coordinates are simulation workspace units rather than meters. The UAVs start with zero velocity near the four corners $(\pm 8, \pm 8)$, with independent uniform coordinate jitter in $[-0.35, 0.35]$. Target coordinates are sampled uniformly from $[-5.8, 5.8]^2$, subject to a minimum target–target separation of 1.5 and a minimum start–target separation of 2.0 workspace units. A target is serviced within radius 0.65 according to the non-exclusive rule in Section 2. Motion uses $\Delta t = 0.2\,\mathrm{s}$, a 40-step horizon, and therefore a maximum simulated duration of 8.0 s. Collision checks use discrete post-step positions, without swept-path checks between updates. A collision is registered below 0.35 workspace units. Mission completion, collision, or a workspace violation terminates an episode; reaching the horizon truncates it.

The original configurations and additional replication runs use the same reward, standard MAPPO update, and 100,000-transition training budget; a separate matched-budget diagnostic also evaluates 200,000 and 300,000 transitions. The parameter-shared actor processes a 13-dimensional ego vector and three 7-dimensional neighbor vectors through 64-unit hidden layers and 32-unit encoders; the training-only central critic processes the 55-dimensional global state through two 128-unit hidden layers. Each rollout contains 512 transitions. Training episodes sample nominal, mild, severe, and dropout events with probabilities 0.25, 0.30, 0.25, and 0.20, respectively; the event step is uniform on $\{8, \ldots, 25\}$, and the affected UAV is sampled uniformly. The final checkpoint after the fixed budget is evaluated deterministically, with no outcome-based checkpoint selection. Table 1 isolates the two locations at which capability information can enter. CA-MAPPO exposes true capability entries to the actor, ER-MAPPO uses them only in event-triggered task allocation, and CAR-MAPPO enables both paths. The historical ER-MAPPO label is retained for the mechanism-selection campaign; this task-layer configuration was subsequently frozen and is referred to as ETCR in the paper and the independent campaign.

For an agent active before the step, the reward is

$$\begin{aligned} r_i = {} & (d_i^- - d_i^+) - 0.01\|\boldsymbol{a}_i - \boldsymbol{a}_i^{\mathrm{prev}}\|^2 - 0.01 \\ & + 8n_i^{\mathrm{service}} + 20I_{\mathrm{complete}} - 25I_{\mathrm{collision}} - 25I_{\mathrm{boundary}}. \end{aligned} \tag{23}$$

The two distances refer to the same pre-motion assigned target (both zero if unassigned); n_i^{service} counts targets credited to agent i. Inactive-agent rewards are zero, and completion and safety terms can coexist. The ego inputs contain actual velocity, previous bounded command, target displacement/distance, remaining-target fraction, own completion count, remaining-time fraction, and capability entries. Neighbor inputs contain relative position/velocity and capability entries. MAPPO and ETCR use nominal capability placeholders in the actor; the centralized critic receives true global positions, velocities, capabilities, targets, completion indicators, assignments, counts, and remaining time. Input scaling is fixed; there is no running observation or reward normalization.

Both Adam learning rates are 3×10^{-4}; $\gamma = 0.99$, GAE $\lambda = 0.95$, PPO clip 0.2, entropy weight 0.01, four update epochs, minibatches of 256 valid agent samples, and gradient-norm limit 0.5 are unchanged. Valid rollout advantages are standardized. Supplementary Section S1 gives the full input packing, networks, reward semantics, RNG and truncation rules, and reproducibility instructions.

The comparative study is designed to compare where capability information enters this fixed hierarchical MAPPO architecture, rather than to rank ETCR against MRTA paradigms with different planning horizons, communication assumptions, or task-bundle semantics.

Table 1. Ablation configurations for policy- and task-level capability utilization in multi-agent proximal policy optimization (MAPPO).

Method	Capability-Conditioned Actor	Capability-Aware Task Allocation	Role
MAPPO	No	No	Non-adaptive reference
CA-MAPPO	Yes	No	Policy-layer capability use
ER-MAPPO	No	Yes	Task-layer capability use
CAR-MAPPO	Yes	Yes	Policy- and task-layer capability use

ER-MAPPO is the task-layer configuration selected in the mechanism-selection campaign and subsequently frozen as ETCR.

Six immutable evaluation conditions are used. The capability vector after the scheduled event is $[1.0, 1.0, 1]^{\mathsf{T}}$ for nominal, $[0.7, 0.8, 1]^{\mathsf{T}}$ for mild degradation, $[0.4, 0.6, 1]^{\mathsf{T}}$ for severe degradation, and $[0.0, 1.0, 0]^{\mathsf{T}}$ for dropout. These four conditions apply the event at step 15. Unseen severity applies $[0.55, 0.70, 1]^{\mathsf{T}}$ at step 15, which is not one of the discrete degradation levels used for training. Unseen timing applies the severe vector at step 5, before the training-time event window. Within each condition, the affected UAV cycles with case index modulo four. Table 2 consolidates the mission, capability, ETCR, training, and evaluation settings; full optimizer and implementation details are provided in Supplementary Section S1.

The first campaign compares MAPPO, CA-MAPPO, ER-MAPPO, and CAR-MAPPO using training seeds 261001–261005. Every trained policy is evaluated on the same 200 fixed cases per condition, giving $4 \times 5 \times 6 \times 200 = 24{,}000$ episodes. This campaign supports architectural ablation, mechanism attribution, and method selection; it is not independent confirmatory evidence, and ETCR was not prespecified before these outcomes were observed. The second campaign compares MAPPO with the frozen ER-MAPPO implementation under the paper name ETCR. It uses new training seeds 262001–262005 and a new 200-case bank per condition, giving $2 \times 5 \times 6 \times 200 = 12{,}000$ episodes. The formal and confirmatory training seeds and case realizations do not overlap, and the two campaigns are analyzed separately.

Table 2. Frozen mission, capability, method, and evaluation settings.

Group	Setting	Frozen Value
Mission	Team and workspace	4 UAVs; 6 targets; abstract planar workspace $[-10, 10]^2$
	Time discretization	$\Delta t = 0.2$ s; 40 steps; maximum duration 8.0 s
	Service and safety	Service radius 0.65; collision distance 0.35 workspace units
	Nominal motion	$\bar{v} = 2.0$ workspace units/s; $\bar{\tau} = 0.5$ s; $\epsilon_h = 0.05$
Capability	Nominal/mild/severe	$(k^v, k^r, \chi) = (1.0, 1.0, 1)/(0.7, 0.8, 1)/(0.4, 0.6, 1)$ at step 15
	Dropout	$(k^v, k^r, \chi) = (0.0, 1.0, 0)$ at step 15
	Unseen cases	Severity $(0.55, 0.70, 1)$ at step 15; timing $(0.4, 0.6, 1)$ at step 5
ETCR	Allocator constants	$\delta_h = 0.15$; $\lambda_s = 0.5$ s; $\lambda_l = 0.75$ s; $\epsilon_v = 10^{-6}$
Training	Budget and rollout	100,000 transitions/run; rollout length 512; final checkpoint
	Event randomization	Nominal/mild/severe/dropout probabilities 0.25/0.30/0.25/0.20; event step uniform on $\{8, \ldots, 25\}$
Evaluation	Mechanism selection	4 methods; 5 seeds; 6 conditions; 200 fixed cases/cell; 24,000 episodes
	Independent confirmation	2 methods; 5 new seeds; 6 conditions; 200 new fixed cases/cell; 12,000 episodes
	Primary confirmation	Severe condition; mission success; paired seed-level bootstrap with 10,000 resamples

Mission success is the fraction of episodes in which all six targets are completed, even if a safety flag occurs at the same terminal step. Safety-qualified completion is reported separately and requires no collision or workspace flag; it does not imply continuous-time collision freedom. Task completion ratio is the completed-target fraction at episode end. Mission completion time is the elapsed time to full completion, and post-event completion time subtracts the scheduled event time; both timing metrics are averaged only over successful episodes and therefore are not unconditional speed measures. Workload imbalance is the sample variance of credited target counts among the four mission participants. Task-switch count sums, over all reassessments, UAV-level changes for which both the previous and new assigned targets are valid. The mechanism audit additionally records whether the capability event triggered reassessment, whether that immediate reassessment changed the assigned UAV for any remaining target, and whether any post-event ETCR assignment later diverged from the distance-only assignment computed from the same state.

The mechanism-selection campaign is summarized descriptively after averaging the 200 cases within each method–seed–condition cell. For the independent campaign, the primary condition is severe degradation, the primary endpoint is mission success, and the inferential unit is the training seed. The protocol first forms five paired ETCR-minus-MAPPO seed-level differences and then computes a two-sided 95% equal-tailed percentile interval from 10,000 paired bootstrap resamples using the frozen random-number seed 261999. Thus, the 1000 evaluation episodes per method and condition comprise five independently trained policies evaluated on 200 fixed cases each; individual episodes are not treated as independent training replicates. Other conditions and metrics retain their prespecified secondary or boundary roles, and the original mechanism-selection and confirmatory case-bank results are not pooled.

4.2. Ablation Study on Capability Adaptation

The first question is whether capability information is more effective in the motion-policy input or in the task-allocation layer. Table 3 reports mission success for the four configurations over five independently trained seeds and 200 fixed cases per seed and condition. These data constitute a two-layer architectural ablation used for mechanism selection; the comparisons are descriptive and are not assigned a confirmatory statistical interpretation.

Table 3. Mission success rates in the capability-adaptation ablation study. Values are five-seed means.

Method	Nominal	Mild	Severe	Dropout	Unseen Severity	Unseen Timing
MAPPO	99.7%	99.3%	88.9%	83.6%	97.9%	79.1%
CA-MAPPO	95.8%	93.2%	81.2%	69.8%	91.9%	72.1%
ER-MAPPO	100.0%	99.7%	98.0%	81.9%	99.5%	92.6%
CAR-MAPPO	97.9%	98.9%	94.8%	78.0%	98.4%	90.5%

At the original 100 k training budget, actor conditioning alone did not improve mean resilience in this setting. Under severe degradation, CA-MAPPO reached 81.2% success compared with 88.9% for MAPPO, a difference of −7.7 percentage points (pp). Adding task-level adaptation instead raised success to 98.0% for ER-MAPPO, or +9.1 pp over MAPPO. The corresponding comparison when actor conditioning was present was also positive: CAR-MAPPO exceeded CA-MAPPO by 13.6 pp. The combined configuration nevertheless remained 3.2 pp below ER-MAPPO. The same pattern persisted under unseen event timing: ER-MAPPO exceeded MAPPO by 13.5 pp, and CAR-MAPPO exceeded CA-MAPPO by 18.4 pp, whereas CA-MAPPO was 7.0 pp below MAPPO and CAR-MAPPO was 2.1 pp below ER-MAPPO. Within this original campaign, task-layer contrasts are positive. The fresh matched-budget results in Section 4.6 substantially change the actor-only comparison; these historical data cannot establish an intrinsic architecture ranking or synergy.

The seed-level view in Figure 3 retains the two genuine learning failures identified in the frozen audit. CA-MAPPO seed 261001 attained 63.5% severe success and 60.0% under unseen timing, while CAR-MAPPO seed 261003 attained 85.0% and 82.5%, respectively. Removing these runs would conceal the larger seed sensitivity of the actor-conditioned variants. By contrast, ER-MAPPO remained between 97.5% and 99.0% in severe degradation and between 90.5% and 94.5% under unseen timing. This stability, rather than a post hoc choice of a favorable seed, motivated freezing the task-layer-only configuration as ETCR.

Dropout provides a boundary rather than evidence for partial-degradation adaptation. ER-MAPPO obtained 81.9% dropout success versus 83.6% for MAPPO. The baseline allocator already releases the failed UAV's task and reallocates among the surviving nominal UAVs, so ETCR is not expected to confer a hard-failure advantage. Across all episodes, task completion ratio followed the severe and unseen-timing success ordering but with less separation: ER-MAPPO versus MAPPO was 99.67% versus 98.15% under severe degradation and 98.70% versus 96.47% under unseen timing. Success-conditional completion times were nearly equal in these ablation comparisons; they therefore do not support an unconditional speed claim.

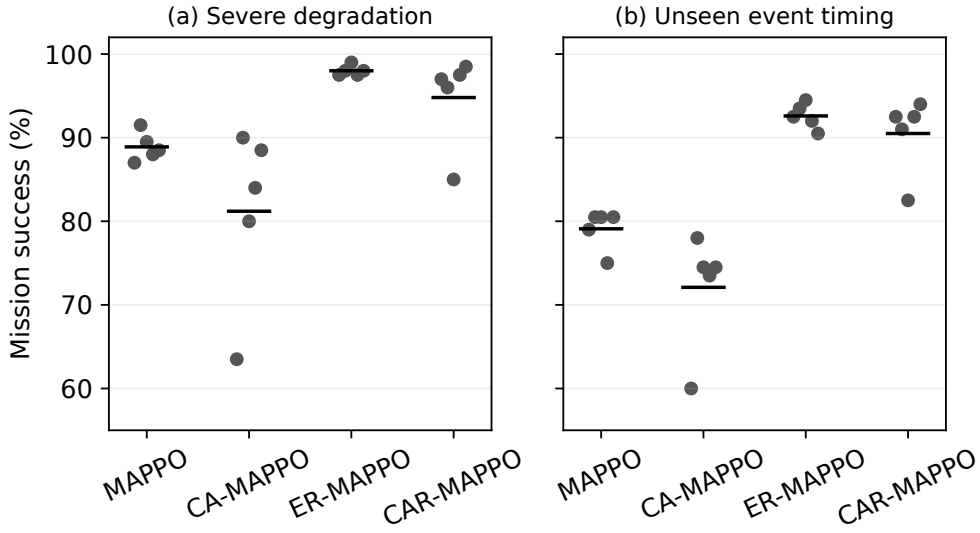


Figure 3. Seed-level mission success in the mechanism-selection campaign under (**a**) severe degradation and (**b**) unseen event timing. Each point denotes one independently trained seed evaluated on 200 fixed cases; short horizontal lines denote five-seed means. All frozen seeds, including the identified CA-MAPPO and CAR-MAPPO learning failures, are retained.

The descriptive mechanism audit clarifies how the selected task-layer configuration acts. It uses the independent confirmatory case bank and is kept separate from the ablation performance aggregates. As summarized in Figure 4, every scheduled mild and severe capability event invoked reassessment, but only 0.7% and 4.1% of episodes, respectively, changed the assignment immediately. Later in the mission, an ETCR assignment differed from the distance-only assignment computed from the same state in 35.7% of mild episodes and 53.2% of severe episodes. Subsequent divergence is not a reassignment rate; it records whether any later capability-aware assignment differs from that same-state reference. Together, these measures are consistent with the intended selective mechanism: a meaningful capability change initiates reassessment, the switching term preserves the current assignment when it remains preferable, and the team restructures later when updated capability and mission geometry justify a different assignment. No communication- or computation-reduction conclusion is drawn.

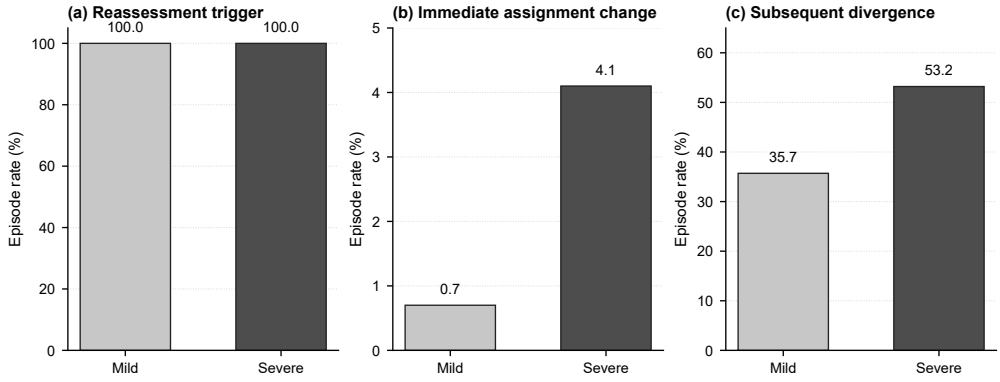


Figure 4. Descriptive ETCR mechanism measures on independent confirmatory cases under mild and severe degradation: (**a**) capability events invoking task reassessment, (**b**) episodes in which the immediate reassessment changed the assignment, and (**c**) episodes with a later ETCR assignment differing from the same-state distance-only reference. Each bar aggregates five trained policies evaluated on 200 cases each.

4.3. Independent Confirmatory Evaluation

After the mechanism-selection campaign, the task-layer-only implementation was frozen as ETCR without outcome-dependent retuning. The independent campaign used the unchanged method and training hyperparameters, new training seeds, and a new fixed evaluation bank with no seed or case overlap with the ablation campaign. Table 4 reports the protocol-frozen seed-level comparison; severe degradation is the sole primary condition.

Table 4. Independent confirmatory results for MAPPO and ETCR across five training seeds.

Condition	MAPPO Mean ± SD	ETCR Mean ± SD	Paired Effect	95% CI
Nominal	98.80 ± 1.40%	99.60 ± 0.42%	+0.80 pp	[−0.40, +2.30] pp
Mild	98.80 ± 0.57%	98.80 ± 1.30%	+0.00 pp	[−1.50, +0.90] pp
Severe	92.90 ± 1.67%	98.00 ± 1.37%	+5.10 pp	[+2.80, +6.60] pp
Dropout	78.60 ± 7.68%	78.50 ± 4.36%	−0.10 pp	[−5.70, +6.60] pp
Unseen severity	96.40 ± 0.96%	97.90 ± 1.08%	+1.50 pp	[+0.60, +2.60] pp
Unseen timing	83.10 ± 2.70%	93.60 ± 1.39%	+10.50 pp	[+8.20, +13.60] pp

SD, standard deviation; CI, confidence interval; pp, percentage points.

Under severe degradation, MAPPO achieved 92.90 ± 1.67% mission success and ETCR achieved 98.00 ± 1.37%. The paired effect was +5.10 pp, with a 95% interval of [+2.80,+6.60] pp. Because the interval for the five seed-level paired differences lies entirely above zero, the protocol-frozen primary criterion was satisfied. All five seed pairs favored ETCR, although the effects ranged from +0.5 to +7.0 pp (Figure 5). The independently estimated severe effect is smaller than the +9.1 pp descriptive estimate in the mechanism-selection campaign, but its direction remains positive after the method and analysis were frozen.

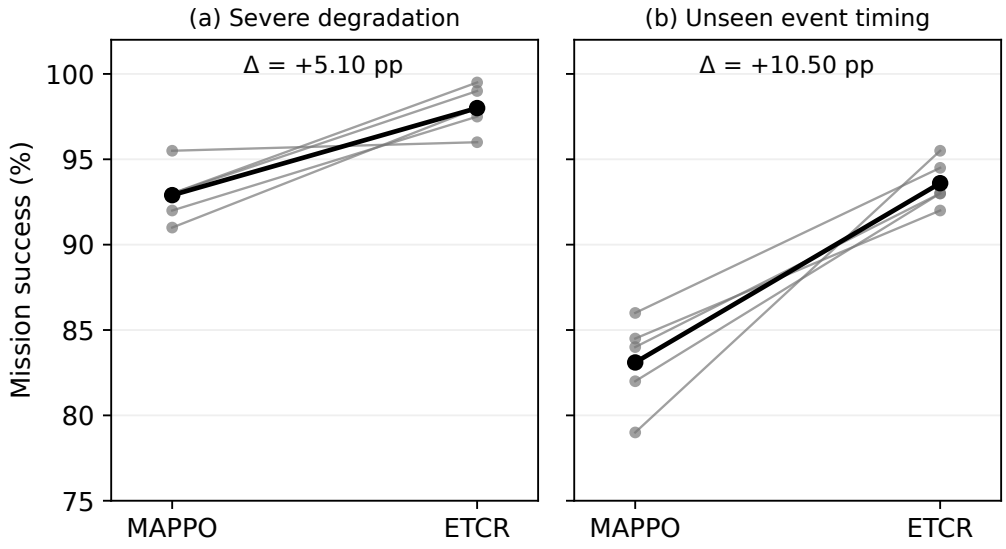


Figure 5. Paired seed-level mission success in the independent confirmatory campaign under (**a**) severe degradation and (**b**) unseen event timing. Each thin line connects MAPPO and ETCR policies with the same confirmatory training-seed identifier, evaluated on the same fixed 200-case bank; the larger connected markers denote five-seed means.

The held-out timing condition supplies the strongest secondary generalization evidence. When the severe event occurred at step 5, outside the training-time event window of steps 8–25, success increased from 83.10 ± 2.70% for MAPPO to 93.60 ± 1.39% for ETCR. The paired effect was +10.50 pp with a 95% interval of [+8.20,+13.60] pp, and every paired

seed improved. This result supports temporal generalization to the tested held-out timing; it does not establish generalization to arbitrary event times. Under unseen severity, both methods retained high success and left limited headroom for separation: ETCR exceeded MAPPO by +1.50 pp, with a 95% interval of [+0.60,+2.60] pp. The mild condition was at ceiling for both methods, with a 0.00 pp paired effect and an interval spanning zero.

Under nominal conditions, ETCR and MAPPO achieved success rates of 99.6% and 98.8%, respectively (paired difference: +0.8 pp; 95% bootstrap CI: [−0.4, +2.3] pp). Under hard dropout, the corresponding rates were 78.5% and 78.6% (−0.1 pp; [−5.7, +6.6] pp). Neither comparison demonstrated a clear difference, and the intervals do not establish equivalence. A hard-dropout advantage therefore remains unestablished.

Figure 6 illustrates execution of the same frozen severe-degradation case by the paired MAPPO and ETCR policies. MAPPO completed five targets but left g_1 unserved at the 8.0 s horizon, with no collision or workspace violation, whereas ETCR completed all six targets at 6.4 s. ETCR's two subsequent allocation divergences from the same-state distance-only reference, together with the observed trajectory differences, are consistent with task-level capability adaptation. This deterministically selected single case is illustrative only and provides no statistical inference, method-selection evidence, or general claim about trajectory speed, path length, or workload.

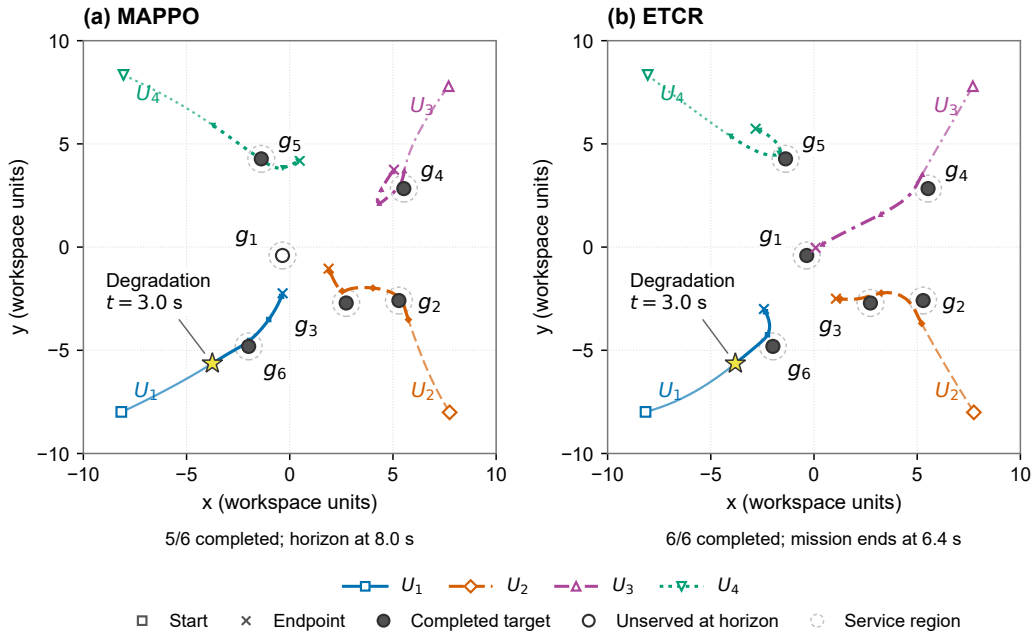


Figure 6. Qualitative mission trajectories under severe capability degradation for a deterministically selected illustrative case from the independent confirmatory campaign. MAPPO and ETCR use the same frozen mission realization and paired training-seed identifier. Seed 262003 has the median paired severe mission-success effect; case index 44 (zero-based) is the lowest index with MAPPO horizon failure without collision or workspace violation, ETCR success, and subsequent ETCR allocation divergence from the same-state distance-only reference. Stars mark the degradation of U_1 at step 15 (3.0 s) to $\boldsymbol{h}_1 = [0.4, 0.6, 1]^\top$. Filled target circles indicate completion; the hollow circle denotes a target unserved at the MAPPO horizon. Dashed rings around targets denote the service radius $r_s = 0.65$ workspace units. Open UAV symbols mark starts and small crosses mark endpoints; lighter trajectory segments indicate pre-degradation motion and darker segments indicate post-degradation motion. This case illustrates mission-level behavior only and is not used for statistical inference or method selection.

Secondary outcomes were used only to check whether the mission-success interpretation was accompanied by coherent mission execution. In severe degradation, task completion ratio increased from 98.80% to 99.67%, while success-conditional mission completion

time decreased from 6.55 to 6.37 s. Under unseen timing, task completion ratio increased from 97.18% to 98.88%, whereas the success-conditional completion times were similar (6.86 and 6.82 s). Because the timing means are conditioned on different successful-episode subsets, they are not interpreted as unconditional efficiency gains. Workload imbalance and task-switch count changed in different directions across conditions, so neither is claimed as a general improvement. The independent results therefore reproduce the central qualitative conclusion of the ablation study: explicit task-level adaptation improves resilience to partial capability degradation in this hierarchical setting, while no clear hard-dropout difference was demonstrated.

4.4. Common-Policy Mechanism Diagnostics

We subsequently evaluated seven allocators with each of the ten unchanged confirmatory actors: five MAPPO-origin and five ETCR-origin actors. Each actor used 200 original cases in nominal, severe, and unseen-timing conditions (42,000 episodes). A0 uses distance divided by nominal speed, no penalties, and no soft capability trigger; A1 adds the original penalties and trigger to that distance cost; A2 is the original ETCR; A3 removes only the soft trigger; A4 removes only the switching cost; A5 removes only workload cost; and A6 replaces the sum objective with a fixed-cardinality bottleneck objective. A6 breaks ties by sum cost, then a fixed index order. These are frozen-policy diagnostics, not independent confirmation, and the two actor origins remain separate. Table 5 summarizes the paired severe-success effects for these contrasts.

Table 5. Paired severe-success effects (pp) with common frozen actors. Positive values favor the first allocator. Origins are separate five-seed groups. Full seed-level results and diagnostic intervals are in Supplementary Sections S3 and S4.

Contrast	MAPPO-Origin	ETCR-Origin
A2–A0: complete ETCR	4.3	4.1
A2–A1: speed-cost component	6.2	5.9
A2–A3: soft trigger	0.0	−0.1
A2–A4: switch penalty	−0.3	0.3
A2–A5: workload penalty	0.5	0.2

The A2–A1 contrast holds penalties and triggers constant and supports a direct contribution from speed-aware assignment under both actor origins. It must not be conflated with A2–A0, which changes the complete allocation mechanism. Removing the soft trigger did not demonstrate an additional success benefit in these step-event cases, although the trigger still invoked capability-event reassessment. Event detection and performance contribution are distinct claims. The workload term is a row constant: if all active UAV rows are assigned, its sum cannot affect the minimum-sum optimum (35,902 numerical checks); when some rows are idle, agent selection can change. The complete A4/A5 results are retained in Supplementary Section S3.

An additional 8000 episodes changed only the severe-event after-vector to speed-only $(0.4, 1)$ or response-only $(1, 0.6)$ at step 15, preserving the original geometry. A2–A0 was +4.8/+4.9 pp for speed-only under MAPPO/ETCR actor origins. Response-only effects were +0.5/+0.7 pp with paired-t intervals crossing zero. In 2000 response-only A2 events, the cost matrix and immediate assignment did not change directly; response dynamics can alter later geometry. Thus, direct adaptation is supported for speed-related degradation, while a stable response-only advantage has not been established.

A6 improved the current matching bottleneck but did not consistently improve whole-mission success. Under unseen timing, A6–A2 was −1.4/−1.5 pp for the two origins.

Current maximum edge cost, maximum selected travel proxy, and sum cost were recorded separately from completion, switches, and horizon failures (Supplementary Section S4). These results support retaining the original sum surrogate in this tested setting; they do not establish that summed cost is a globally correct makespan objective or rule out better long-horizon scheduling.

4.5. Revision-Stage Seed Extension and Statistical Sensitivity

The original five severe differences were [7.0, 0.5, 6.0, 6.5, 5.5] pp. The original +5.10 pp effect retains its frozen bootstrap interval [2.80,6.60] pp; a paired-t interval is [1.83,8.37] pp, and the two-sided sign/sign-flip result is $p = 0.0625$. These distinct summaries expose the resolution limit of five pairs. Supplementary Section S2 reports the original paired differences and statistical sensitivity. The original endpoint counts completion even if a collision occurs at the same endpoint. Safety-qualified completion, requiring no collision or boundary flag, gives a separate +5.00 pp severe effect, not a redefinition of the +5.10 pp primary endpoint.

During revision, five new paired seeds (263001–263005) were trained at the unchanged 100,000-transition budget, with final-checkpoint evaluation on all six original conditions. A second 1200-case bank (200 per condition) was generated and hashed before evaluation; all original and new actors were evaluated on that same new bank. The new bootstrap (10,000 paired-seed resamples), paired-t intervals, exact sign test, and sign-flip sensitivity were fixed before running. The unit remains the training seed, not the individual case. This is a revision-stage extension, not a claim of preregistration before the original submission. Table 6 summarizes the severe-success replication across cohorts and case banks.

Table 6. Severe-success replication. Effects and intervals are pp, ETCR minus MAPPO. Bootstrap is the prespecified percentile procedure for each campaign.

Cohort/Case Bank	*n*	Effect	Bootstrap 95%	Paired-t 95%
Original/original	5	5.10	[2.80, 6.60]	[1.83, 8.37]
New/original	5	7.70	[5.70, 9.40]	[4.77, 10.63]
Original/new	5	8.00	[5.20, 10.90]	[3.31, 12.69]
New/new	5	10.10	[8.50, 11.70]	[7.53, 12.67]
Combined/new	10	9.05	[7.20, 10.80]	[6.85, 11.25]

All ten severe differences on the new bank were positive (two-sided sign $p = 0.001953$). The larger effect cannot be attributed to sample size alone: the case realization and training numerical runtime also differ. Intervals describe across-training-seed variation conditional on each fixed bank. The original five-seed study is preserved separately, and complete per-seed/all-condition results appear in Supplementary Section S5. On the new bank, the severe horizon-failure effect was −9.05 pp; collision and boundary effects were −0.10 and −0.05 pp, respectively. Completion–collision overlap changed by −0.15 pp, giving a separately labeled safety-qualified effect of +9.20 pp. Failure flags overlap and are not a mutually exclusive partition. Success-conditional time was lower by 0.140 s, but different successful subsets prevent an unconditional efficiency interpretation.

Table 7 reports the original confirmation secondary outcomes for severe degradation and unseen timing.

Table 7. Original confirmation secondary outcomes, mean ± sample SD over five seeds. Time includes successful episodes only; workload is the sample variance of completed-task counts.

Condition	Method	Time (s)	Workload	Switches	Horizon (%)
Severe	MAPPO	6.55 ± 0.11	0.64 ± 0.02	3.20 ± 0.05	6.80 ± 1.75
Severe	ETCR	6.37 ± 0.10	0.65 ± 0.02	3.26 ± 0.05	1.70 ± 1.30
Timing	MAPPO	6.86 ± 0.11	0.84 ± 0.03	3.51 ± 0.03	16.60 ± 2.88
Timing	ETCR	6.82 ± 0.08	0.87 ± 0.02	3.41 ± 0.04	5.90 ± 1.39

4.6. Matched Training-Budget Diagnostic

We retrained MAPPO and CA-MAPPO from scratch for every formal seed (261001–261005), saving all 100 k, 200 k, and 300 k checkpoints without selecting by validation performance. Every checkpoint was evaluated using the same formal nominal, severe, and unseen-timing banks. Each 100 k segment ends with the original partial rollout, then continues with the in-memory optimizer and sampling states; both methods have identical segment boundaries. The new 100 k prefixes did not exactly reproduce the historical checkpoints. A 512-step diagnostic reproduced the historical first update using 12 CPU threads, whereas the locked single-thread runs differed initially by about 10^{-8}. This identifies the first numerical discrepancy, not proof of full-run reproduction. Consequently, this experiment is labeled matched-budget fresh retraining, not seamless extension of the historical models. Table 8 reports severe success at each matched training budget.

Table 8. Severe success (mean ± sample SD, %) for all five matched-budget runs. Every checkpoint is retained.

Method	100 k	200 k	300 k
MAPPO	88.5 ± 0.71	90.2 ± 2.14	89.9 ± 2.56
CA-MAPPO	88.8 ± 1.40	88.6 ± 2.77	87.8 ± 2.17

CA-MAPPO did not improve monotonically with added budget. However, the historical actor-only deficit was not reproduced at the new 100 k point. In particular, seed 261001 reached 88.5% severe success at that point, so its recovery cannot be credited to extended training. Paired-t intervals for CA-MAPPO minus MAPPO cross zero at each budget. We therefore retain the historical outcomes but withdraw an intrinsic architecture ranking: they depend on the training realization and numerical setting. The independent ETCR evidence and common-policy allocator contrasts address different questions. Supplementary Section S6 separates per-seed training traces from fixed-case held-out evaluation curves.

4.7. Limited Sensitivity and Generalization

A further prespecified 14,000-episode diagnostic used all ten original actors and the first 50 cases per severe/timing bank, with actor origins kept separate. The switch weight was varied over 0.25/0.5/1.0 s and the workload weight over 0/0.75/1.5 s, one factor at a time. These are sensitivity checks of frozen engineering settings, not optimization of the primary parameters; no globally optimal choice is claimed. The original setting was retained. Raising switching cost to 1.0 s reduced timing success in the small subset, illustrating that assignment continuity can become too restrictive.

Thresholds 0.1/0.15/0.3 were also tested on step changes of 0.12 or 0.18 in speed ratio, with response unchanged. Reassessment followed the strict threshold, but no success difference was observed between these threshold choices in the small subset. This does not prove threshold invariance or cumulative-degradation detection. At 32/40/50 steps, Δt and geometry were fixed, but the actor's original $1 - t/H$ feature changed with the horizon.

Shortening to 32 substantially reduced success; these coupled deadline/encoding results do not establish arbitrary horizon generalization.

Estimate-only Gaussian noise (SD 0.05, clipped capability ratios) and a two-step estimate delay left true physical capability unchanged. The noise produced additional capability reassessments (about 3.6 per severe episode and 5.1 under unseen timing versus 1.0 with exact estimates), without a large mean success change in this subset. Such extra triggers are a practical limitation, not evidence for communication savings. Diagnostic protocols, summaries across the five seeds per actor origin, and full/partial-row allocation diagnostics appear in Supplementary Section S7. These limited tests do not support larger UAV/target populations or a claim of real-flight robustness.

5. Conclusions

ETCR updates current task guidance using speed-related execution capability while retaining a standard MAPPO motion policy. In the tested four-UAV, six-target missions, the original severe-degradation gain of +5.10 pp was supported by five additional training pairs (+7.70 pp on the original bank). All ten pairs on a new fixed bank yielded +9.05 pp. Common frozen-policy comparisons support a direct contribution from the speed-aware allocation cost under both actor origins, distinct from the effects of separately trained systems.

The event-triggered scheduler invokes reassessment without forcing an assignment change. Its soft capability trigger did not show an additional success benefit in the tested step-event cases. Speed-only benefits were supported, whereas stable response-only and hard-dropout advantages were not established. Improving the current matching bottleneck with minmax allocation did not consistently improve full-mission success. These results support retaining the original receding sum-cost allocator within this scope, without claiming makespan optimality or an independent benefit from every component.

Matched-budget fresh retraining did not reproduce the historical CA-MAPPO deficit at 100 k and showed no monotonic improvement through 300 k. The evidence therefore supports task-level speed adaptation without establishing an intrinsic ranking of actor conditioning and task-layer adaptation.

The model limits commanded and steady-state speed, permits transient actual-speed overshoot after a capability reduction, and uses a travel-cost proxy rather than exact dynamic arrival time. The study assumes centralized task information and externally supplied capabilities. Endpoint safety checks, the fixed team and target counts, and the tested event distributions limit the scope of inference. The one-step trigger does not guarantee detection of cumulative subthreshold changes; limited-horizon and estimate-error diagnostics do not establish general deployment robustness. Response-aware costs, cumulative event detection and longer-horizon planning remain directions for separately validated extensions.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/machines14111326/s1. Table S1: Frozen environment and execution parameters; Table S2: Actor ego inputs and deterministic scaling; Table S3: Actor neighbor inputs and deterministic scaling; Table S4: Network-training and PPO optimization settings; Table S5: Original five-pair mission-success differences and interval sensitivity. Effects and intervals are in percentage points (ETCR minus MAPPO). Panel A lists the individual differences; Panel B compares interval procedures and the two-sided sign-flip test; Table S6: Common-policy outcomes, mappo actor origin, n = 5; Table S7: All five training-seed success rates (%), mappo origin; Table S8: All prespecified diagnostic success contrasts, mappo origin; Table S9: Common-policy outcomes, er_mappo actor origin, n = 5; Table S10: All five training-seed success rates (%), er_mappo origin; Table S11: All prespecified diagnostic success contrasts, er_mappo origin; Table S12: Task completion (%); five seeds per actor origin; Table S13: Success-conditional completion time (s); five seeds per actor origin; Table S14: Workload imbalance (sample variance of completed counts); five seeds per actor origin;

Table S15: Task-switch count; five seeds per actor origin; Table S16: Horizon failure (%); five seeds per actor origin; Table S17: Collision flag (%); five seeds per actor origin; Table S18: Boundary flag (%); five seeds per actor origin; Table S19: Completion and collision overlap (%); five seeds per actor origin; Table S20: Safety-qualified completion (%); five seeds per actor origin; Table S21: Every isolated-capability seed success rate (percent); Table S22: Means of seed-level allocation summaries (seconds). A2/A6 visit different states; only the final two columns compare solvers on the same state; Table S23: Severe paired-seed statistical sensitivity; Table S24: Original-bank per-seed success (%): nominal, mild and severe conditions. ETCR is ER-MAPPO in code; Table S25: Original-bank per-seed success (%): dropout, unseen severity and unseen timing; Table S26: New-bank per-seed success (%): nominal, mild and severe conditions; original and new training cohorts; Table S27: New-bank per-seed success (%): dropout, unseen severity and unseen timing; original and new training cohorts; Table S28: All-condition expanded ten-pair effects; non-severe outcomes descriptive; Table S29: Matched-budget means and sample SD across five seeds; Table S30: All checkpoint success rates (%), with no seed or checkpoint filtering; Table S31: All small-diagnostic results, mappo origin; Table S32: All small-diagnostic results, er mappo origin; Table S33: Estimate-trigger diagnostics; empty means no qualifying trigger; Table S34: Allocation phases: full means all active UAV rows selected; partial means some active UAVs idle. Baseline phase fields were recovered from the unchanged allocation records; all values are means per episode across the five seeds; Figure S1: All ten severe paired differences on the same new case bank. Gray bars identify original training seeds and teal bars identify new training seeds; original-bank differences are tabulated separately; Figure S2: Training rollout recent-success traces, ten-update smoothing, all seeds. These are sampled training episodes, not fixed-bank held-out scores; Figure S3: Fixed held-out evaluation at every prespecified budget, all five seeds and both methods; separate from training curves.

Author Contributions: Conceptualization, Q.Y.; methodology, Q.Y.; software, Z.X.; validation, K.Y.; formal analysis, K.Y.; investigation, K.Y.; data curation, Z.X.; writing—original draft preparation, Z.X.; writing—review and editing, Z.X.; supervision, Y.Z. and S.S.; project administration, Q.Y., Y.Z. and S.S.; funding acquisition, Q.Y. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61903324; the Natural Science Foundation of Shandong Province of China, grant number ZR2017LF013; the Science and Technology Development Plan of Zaozhuang (Industrial Application Technology), grant number 2023GX07; and the Initiative for Enhancing the Innovation Capability of Technology-based SMEs of Shandong Province, China, grant number 2025TSGCCZZB0408.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article/Supplementary Material. The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used OpenAI ChatGPT (GPT-5.6) and Codex (GPT-5.6 Sol) for language refinement, manuscript formatting, code-assisted figure preparation, and technical consistency checking. The authors reviewed and edited the outputs and take full responsibility for the content of this publication.

Conflicts of Interest: Q.Y. has a collaborative relationship with Jinzhong Hydraulic Rubber Hose Co., Ltd., which did not provide financial support for this study. The other authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CA-MAPPO	Capability-conditioned actor MAPPO variant
ER-MAPPO	Event-triggered task-reallocation MAPPO variant
CAR-MAPPO	Combined actor-and-reallocation MAPPO variant
ETA	Speed-based travel-cost proxy (historical ETA notation)
ETCR	Event-Triggered Capability-Aware Task Reallocation
MAPPO	Multi-Agent Proximal Policy Optimization
MRTA	Multi-Robot Task Allocation
UAV	Unmanned Aerial Vehicle

References

1. Gerkey, B.P.; Matarić, M.J. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [CrossRef]
2. Korsah, G.A.; Stentz, A.; Dias, M.B. A Comprehensive Taxonomy for Multi-Robot Task Allocation. *Int. J. Robot. Res.* **2013**, *32*, 1495–1512. [CrossRef]
3. Alqefari, S.; Menai, M.E.B. Multi-UAV Task Assignment in Dynamic Environments: Current Trends and Future Directions. *Drones* **2025**, *9*, 75. [CrossRef]
4. Dias, M.B.; Zlot, R.; Kalra, N.; Stentz, A. Market-Based Multirobot Coordination: A Survey and Analysis. *Proc. IEEE* **2006**, *94*, 1257–1270. [CrossRef]
5. Choi, H.L.; Brunet, L.; How, J.P. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926. [CrossRef]
6. Ferreira, B.A.; Petrović, T.; Orsag, M.; Martínez-de Dios, J.R.; Bogdan, S. Distributed Allocation and Scheduling of Tasks with Cross-Schedule Dependencies for Heterogeneous Multi-Robot Teams. *IEEE Access* **2024**, *12*, 74327–74342. [CrossRef]
7. Heppner, G.; Oberacker, D.; Roennau, A.; Dillmann, R. Behavior Tree Capabilities for Dynamic Multi-Robot Task Allocation with Heterogeneous Robot Teams. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 4826–4833. [CrossRef]
8. Bischoff, E.; Kohn, S.; Hahn, D.; Braun, C.; Rothfuß, S.; Hohmann, S. Heuristic reoptimization of time-extended multi-robot task allocation problems. *Networks* **2024**, *84*, 64–83. [CrossRef]
9. Verma, A.; Gautam, A.; Shekhawat, V.S.; Mohan, S. DTA-HMR-TT: Dynamic Task Allocation for a Heterogeneous Team of Mobile Robots with Task Transfer. *IEEE Access* **2024**, *12*, 177050–177063. [CrossRef]
10. Mayya, S.; D'Antonio, D.S.; Saldaña, D.; Kumar, V. Resilient Task Allocation in Heterogeneous Multi-Robot Systems. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1327–1334. [CrossRef]
11. Notomista, G.; Mayya, S.; Emam, Y.; Kroninger, C.; Bohannon, A.; Hutchinson, S.; Egerstedt, M. A Resilient and Energy-Aware Task Allocation Framework for Heterogeneous Multirobot Systems. *IEEE Trans. Robot.* **2022**, *38*, 159–179. [CrossRef]
12. Li, L.; Chen, Z.; Wang, H.; Kan, Z. Task Allocation of Heterogeneous Robots Under Temporal Logic Specifications with Inter-Task Constraints and Variable Capabilities. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 14030–14047. [CrossRef]
13. Calvo, Á.; Capitán, J. Heterogeneous Multirobot Task Allocation for Long-Endurance Missions in Dynamic Scenarios. *IEEE Trans. Robot.* **2025**, *41*, 6494–6513. [CrossRef]
14. Wang, X.; Gao, X.; Wang, L.; Su, X.; Jin, J.; Liu, X.; Deng, Z. Resilient multi-objective mission planning for UAV formation: A unified framework integrating task pre- and re-assignment. *Def. Technol.* **2025**, *45*, 203–226. [CrossRef]
15. Jiang, Z.; Song, T.; Yang, B.; Song, G. Fault-Tolerant Control for Multi-UAV Exploration System via Reinforcement Learning Algorithm. *Aerospace* **2024**, *11*, 372. [CrossRef]
16. Liu, T.; Wei, S.; Liu, C.; Bai, G.; Zhang, J.; Yuan, C. A Resilient Task Allocation Method for Unmanned Aerial Vehicle Swarm Considering Local Information. *Reliab. Eng. Syst. Saf.* **2026**, *272*, 112442. [CrossRef]
17. Zeng, Y.; Wu, L.; Li, J.; Zhuang, X. Enhancing UAV Swarm Resilience to Payload Failures: A Bidirectional Task Reallocation Approach. *IEEE Syst. J.* **2026**, *20*, 223–234. [CrossRef]
18. Cui, Y.; Chen, X.; Huang, W.; Pan, R.; Tian, Y.; Huang, T.; Basin, M.V. STAP^2: Decentralized Spatiotemporal Task Allocation and Path Planning for Time-Sensitive Multi-UAV Missions. *IEEE Trans. Ind. Inform.* 2026, *Early Access*. [CrossRef]
19. Hwang, S.Y.; Oh, S.; Lee, J.C.; Lee, S.; Lee, J.H.; Hwang, H.J. Assignment-Preserving Replanning for Dynamic Task Allocation in Multi-UAV Systems. *Appl. Sci.* **2026**, *16*, 9414. [CrossRef]
20. Yu, J.; Zhang, Y.; Sun, C. Multi-UAV Dynamic Task Assignment Based on Event-Triggered Graph Reinforcement Learning Under Weak Communication. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 19106–19121. [CrossRef]
21. Aznar, F.; Pujol, M.; Díez, Á. Event-Triggered Adaptive Consensus for Multi-Robot Task Allocation. *Comput. Commun.* **2026**, *251*, 108499. [CrossRef]

22. Hu, C.; Li, Y.; Qu, G. Integrated dynamic task allocation via event-triggered for tracking ground moving targets by UAVs in urban. *Robot. Auton. Syst.* **2025**, *193*, 105061. [CrossRef]
23. Motes, J.; Sandström, R.; Lee, H.; Thomas, S.; Amato, N.M. Multi-Robot Task and Motion Planning with Subtask Dependencies. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3338–3345. [CrossRef]
24. Forte, P.; Mannucci, A.; Andreasson, H.; Pecora, F. Online Task Assignment and Coordination in Multi-Robot Fleets. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4584–4591. [CrossRef]
25. Neville, G.; Chernova, S.; Ravichandar, H. D-ITAGS: A Dynamic Interleaved Approach to Resilient Task Allocation, Scheduling, and Motion Planning. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1037–1044. [CrossRef]
26. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 6379–6390.
27. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
28. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 24611–24624. [CrossRef]
29. Dai, W.; Rai, U.; Chiun, J.; Cao, Y.; Sartoretti, G. Heterogeneous Multi-robot Task Allocation and Scheduling via Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2654–2661. [CrossRef]
30. Yuan, Z.; Wang, R.; Kim, T.; Zhao, D.; Obi, I.; Min, B.C. Adaptive Task Allocation in Multi-Human Multi-Robot Teams Under Team Heterogeneity and Dynamic Information Uncertainty. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025; pp. 1–7. [CrossRef]
31. Wang, W.; Bera, A.; Min, B.C. Hypergraph-Based Coordinated Task Allocation and Socially-Aware Navigation for Multi-Robot Systems. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025; pp. 16225–16232. [CrossRef]
32. Huang, S.; Sun, C.; Gong, J.; Pompili, D. Reinforcement learning-based task allocation and path-finding in multi-robot systems under environment uncertainty. *Comput.-Aided Civ. Infrastruct. Eng.* **2025**, *40*, 3408–3429. [CrossRef]
33. Dimarogonas, D.V.; Frazzoli, E.; Johansson, K.H. Distributed Event-Triggered Control for Multi-Agent Systems. *IEEE Trans. Autom. Control* **2012**, *57*, 1291–1297. [CrossRef]
34. Åström, K.J.; Murray, R.M. *Feedback Systems: An Introduction for Scientists and Engineers*; Princeton University Press: Princeton, NJ, USA, 2008. [CrossRef]
35. Crouse, D.F. On implementing 2D rectangular assignment algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 1679–1696. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.