electronics

MDPI

Article

MARL-Driven Decentralized Crowdsourcing Logistics for Time-Critical Multi-UAV Networks

Juhyeong Han and Hyunbum Kim *

Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, Republic of Korea; hph6489@naver.com

* Correspondence: hyunbumkim@ieee.org; Tel.: +82-32-835-8764

Abstract

Centralized UAV logistics controllers can achieve strong navigation performance in controlled settings, but they do not capture key deployment factors in crowdsourcing-enabled emergency logistics, where heterogeneous UAV owners participate with unreliability and dropout, and incentive expenditure and fairness must be accounted for. This paper presents a decentralized crowdsourcing multi-UAV emergency logistics framework on an edge-orchestrated architecture that (i) performs urgency-aware dispatch under distance/energy/payload constraints, (ii) tracks reliability and participation dynamics under stress (unreliable agents and dropout), and (iii) quantifies incentive feasibility via total payment and payment inequality (Gini). We adopt a hybrid decision design in which PPO/DQN policies provide real-time navigation/control, while GA/ACO act as planning-level route refinement modules (not reinforcement learning) to improve global candidate quality under safety constraints. We evaluate the framework in a controlled grid-world simulator and explicitly report stress-matched re-evaluation results under matched stress settings, where applicable. In the nominal comparison, centralized DQN attains high navigation-centric success (e.g., 0.970 ± 0.095) with short reach steps, but it omits incentives by construction, whereas the proposed crowdsourcing method reports measurable payment and fairness outcomes (e.g., payment and Gini) and remains evaluable under unreliability and dropout sweeps. We further provide a utility decomposition that attributes negative-utility regimes primarily to collision-related costs and secondarily to incentive expenditure, clarifying the operational trade-off between mission value, safety risk, and incentive cost. Overall, the results indicate that navigation-only baselines can appear strong when participation economics are ignored, while a deployable crowdsourcing system must explicitly expose incentive/fairness and robustness characteristics under stress.

Keywords: Multi-Agent Reinforcement Learning (MARL); Unmanned Aerial Vehicles (UAVs); decentralized crowdsourcing; emergency logistics; edge computing; task allocation; path planning; collision avoidance

Check for updates

Academic Editors: Zhiyuan Zhang, Yang Zhang and Zhao Li

Received: 8 December 2025
Revised: 27 December 2025
Accepted: 8 January 2026
Published: 12 January 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.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have become an important enabler for time-critical logistics, including the rapid delivery of medical supplies, food, and emergency equipment in disaster-stricken areas. Prior UAV logistics systems have typically relied on centralized fleet management, fixed resource provisioning, and preconfigured routing or scheduling policies [1,2]. While effective under controlled conditions, these designs often exhibit limited scalability and adaptability when the environment becomes highly

dynamic, e.g., sudden demand surges, partial infrastructure failures, flight restrictions (e.g., "no-drone zones"), and heterogeneous UAV availability [3,4]. In practice, centralized approaches also create single points of failure and bottlenecks that can delay task dispatch and re-planning during emergencies.

A promising direction is *crowdsourcing-enabled multi-UAV emergency logistics* , where UAVs owned by individuals, enterprises, and public agencies can be temporarily integrated into a common operational pool. Such a model can expand capacity on demand, improve geographic coverage, and reduce dependence on a single operator. However, crowdsourcing introduces new challenges: (i) decentralized coordination among heterogeneous UAVs with different battery/energy constraints and reliability, (ii) robust task assignment under time pressure and safety constraints (e.g., collision avoidance), (iii) routing and re-planning under obstacles and flight restrictions, and (iv) participation management with incentives and fairness when privately owned UAVs participate.

To address these challenges, we propose a *crowdsourced multi-UAV emergency logistics framework* that combines learning-based decision policies (PPO and DQN) with planning-level route refinement (GA and ACO), together with an incentive-aware dispatch mechanism. The design goal is not only navigation-centric success, but also *crowdsourcing feasibility under stress*, which requires measuring incentive outcomes (total payment and inequality) and robustness against unreliable participants and participation dropout. Consequently, our evaluation explicitly reports *baseline-under-stress* results under the same stress settings, in addition to nominal-condition comparisons.

Figure 1 illustrates a conventional setting in which a single policy (or a centrally trained controller) handles routing and task decisions. Such a design can be brittle in emergency scenarios, where a single controller must continuously re-optimize under uncertain demand and constraints. In contrast, Figure 2 illustrates a crowdsourcing-enabled setting in which multiple UAVs can be dynamically recruited and coordinated through decentralized mechanisms.

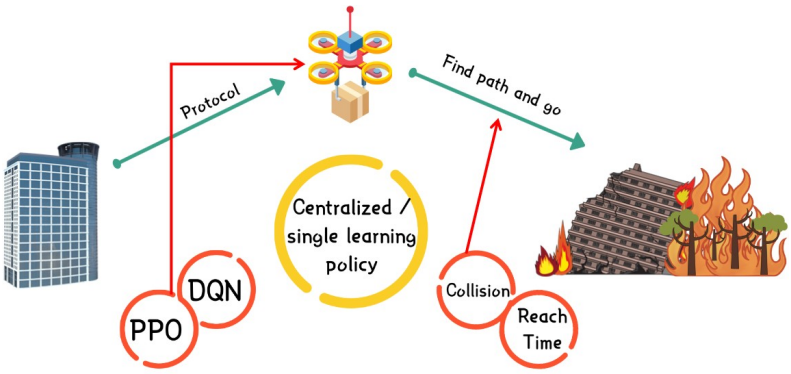


Figure 1. UAV routing under traditional centralized or single-policy learning frameworks.

Why reinforcement learning and what are its limitations?

Reinforcement learning (RL) is attractive for emergency logistics because it can optimize sequential decisions under uncertainty, enabling agents to adapt to stochastic demand and evolving constraints [5–8]. Nevertheless, RL may lack optimality guarantees, can be sensitive to reward design/hyperparameters, and may degrade under environment shifts. Therefore, our framework is designed as a *hybrid* decision architecture: learning-based policies are used for real-time decision-making and navigation, while planning-level refinement (GA/ACO) and explicit safety constraints are incorporated to improve robustness and interpretability. We emphasize that the evaluation is simulation-based; bridging the gap to real-world deployment requires additional modeling of 3D dynamics, communication disruptions, and operational constraints (discussed in the concluding section).

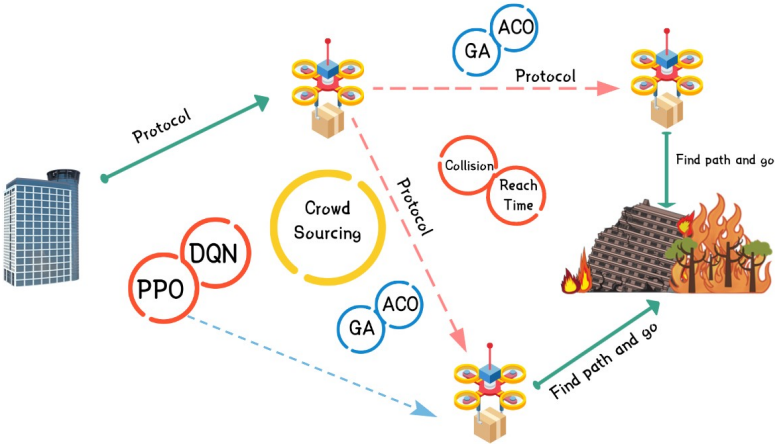


Figure 2. A crowdsourcing-enabled multi-UAV environment that integrates heterogeneous UAV resources and incentive-aware dispatch for emergency missions.

Contributions.

The main contributions of this paper are summarized as follows:

- **Crowdsourced multi-UAV framework with incentive-aware dispatch:** We introduce a crowdsourcing-enabled emergency logistics framework that coordinates heterogeneous UAV participants based on real-time status and reliability-related signals, reducing reliance on fixed fleets.
- **Hybrid decision architecture (policy-level RL + planning-level refinement):** We integrate PPO/DQN-based decision policies with GA/ACO route refinement to improve route quality and robustness in constrained environments [7–11].
- **Incentive and fairness characterization:** Unlike centralized baselines that do not model payments, we quantify total payment and payment inequality (Gini) to evaluate crowdsourcing feasibility.
- **Stress-matched baseline re-evaluation (where applicable):** We re-evaluate centralized PPO/DQN under crowdsourcing-relevant stressors that remain well-defined for baselines (e.g., unreliability as stochastic failure/noise), using identical map generation, obstacle placement, and collision rules. We explicitly note that participation dropout is not defined for centralized baselines and is evaluated only for the crowdsourcing regime.

This paper is organized as follows. Section 2 reviews related work on UAV logistics, decentralized coordination, crowdsourcing, and learning-based decision-making. Section 3 presents the system overview and problem formulation. Section 4 describes the proposed algorithms and operational workflow. Section 5 reports the simulation setup and experimental results (including stress tests and baseline-under-stress). Finally, Section 6 concludes the paper and outlines limitations and future research directions.

2. Related Work

2.1. UAV Logistics and Coordination Paradigms

Prior research on UAV-based logistics has largely focused on centrally managed fleets operated by companies or public agencies. Centralized designs can simplify control and compliance management, but they often face scalability limitations, high operational overhead, and reduced robustness under large-scale disruptions or rapidly changing demand [1,2]. Moreover, their reliance on fixed fleet capacity makes it difficult to elastically scale during emergencies, where additional UAV resources may be urgently required.

A major line of work investigates *swarm and metaheuristic optimization* for multi-UAV coordination and routing. Methods inspired by collective behaviors, including Ant

Colony Optimization (ACO), Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and related heuristics, have been applied to multi-UAV path planning and task coordination [9–11]. These approaches can produce efficient routes and cooperative behaviors, but they are commonly evaluated under assumptions of known objectives and relatively stable environments, and they may still depend on centralized orchestration or predefined behavioral rules when deployed.

Another line of work uses *learning-based navigation and decision-making*. Supervised learning and reinforcement learning have been adopted for UAV navigation, obstacle avoidance, and dynamic scheduling [6,12]. While learning-based approaches offer adaptivity, they introduce practical concerns such as training stability, sensitivity to reward shaping, and generalization under environment shifts. These limitations motivate hybrid designs that integrate learning with explicit planning and safety constraints.

A further research direction explores *decentralized coordination infrastructures*, including blockchain-based mechanisms for secure logging, accountability, and trust management. Such systems can improve integrity and traceability, but they do not by themselves guarantee fast, adaptive decision-making under time-critical constraints. In emergency logistics, latency and re-planning responsiveness remain essential requirements, motivating lightweight edge-enabled orchestration.

2.2. Crowdsourcing-Enabled UAV Resource Expansion

Crowdsourcing UAV logistics aims to expand the available UAV pool by allowing privately owned UAVs to participate in emergency missions under a unified coordination mechanism [13,14]. Compared to fixed-fleet approaches, crowdsourcing can improve scalability and coverage, particularly when public infrastructure is disrupted. However, this paradigm requires solutions for participant management, incentive alignment, and trust assessment (e.g., reputation), as well as mechanisms to integrate heterogeneous UAV constraints (battery, payload, flight capability) into task assignment and routing.

In addition, crowdsourcing introduces challenges in coordination and communication. Inter-UAV information exchange may be intermittent, and decentralized execution must remain robust to partial failures and missing updates. Consequently, practical crowdsourcing frameworks often incorporate edge-level coordination to reduce long-haul dependency and to enable rapid local decision-making in disaster regions.

2.3. Multi-Agent Reinforcement Learning for Task Allocation and Navigation

Multi-Agent Reinforcement Learning (MARL) provides a principled approach to learning cooperative policies for multi-UAV decision-making under uncertainty. Representative MARL formulations include centralized training with decentralized execution (CTDE), which can stabilize learning while enabling distributed deployment [5]. MARL has been investigated for task allocation, coverage control, and coordination in dynamic environments, where multiple agents must share limited resources and avoid conflicts.

In emergency logistics, MARL is attractive because it can learn policies that jointly optimize response time and safety while adapting to stochastic demand patterns. However, MARL also faces known limitations: training may be unstable or computationally expensive as the number of agents grows, learned policies can be sensitive to reward definition, and performance can degrade under distribution shifts. These challenges motivate hybrid architectures in which MARL supports real-time decisions (e.g., dispatching and local navigation) while global route refinement and explicit safety constraints are integrated to improve robustness and interpretability. Our work follows this direction by proposing a decentralized crowdsourcing edge network with cooperative MARL-based decision policies, complemented by planning-level refinement.

3. Crowdsourced Multi-UAV Framework and Utilization of Crowdsourcing

This section presents the proposed *crowdsourced multi-UAV emergency logistics framework* for coordinating heterogeneous UAV resources contributed by individuals, enterprises, and public agencies. We clarify system roles, the decision/planning components, and (critically) the **in-scope vs. out-of-scope** modules for the current evaluation in Section 5 to avoid ambiguity in reviewer-facing interpretation.

3.1. System Overview and Roles of Key Modules

The proposed framework scales emergency logistics capacity by integrating dynamically available UAVs into a unified operational pool. It consists of (i) a crowdsourcing layer to enroll heterogeneous UAVs, (ii) an edge orchestrator to maintain a task pool and aggregate UAV status, and (iii) a decision/planning layer to support real-time dispatch and safe navigation. Figure 3 represents a brief architectural design of the crowdsourced multi-UAV emergency logistics framework.

Crowdsourcing layer (participant enrollment).

UAV owners register UAV profiles including location, residual battery/energy, payload capability, and reliability-related attributes (initialized priors). Importantly, our study models **participation dynamics** that are central to crowdsourcing: UAVs can be unreliable and can drop out. These two stress dimensions are explicitly evaluated in Section 5.

Edge orchestrator (task pool and constraint gatekeeping).

The edge orchestrator maintains (a) a *task pool* containing emergency requests and constraints, (b) *status reports* (location/energy/availability), and (c) a *dispatch channel* for assignments. It enforces operational constraints (e.g., collision safety and restricted zones) and provides bounded state/context information to support decentralized execution.

Decision and planning layer (hybrid design).

We adopt a hybrid decision architecture: PPO/DQN-based policies are used for *real-time navigation/control* (policy level), while GA/ACO are used for *planning-level route candidate generation and refinement*. We emphasize that GA/ACO are **metaheuristic planners**, not reinforcement learning algorithms; they complement learning-based control rather than replacing it.

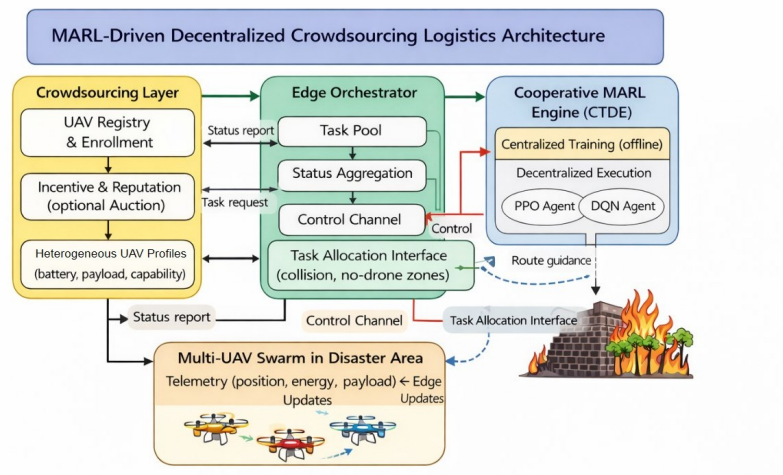


Figure 3. Architectural design of the crowdsourced multi-UAV emergency logistics framework. The edge orchestrator aggregates tasks and UAV status; policy-level RL supports real-time decisions; planning-level GA/ACO refines global route candidates under safety constraints.

3.2. Learning-Based Decisions and Hybrid Planning Rationale

RL is suitable for emergency logistics because it supports sequential decisions under uncertainty. In this paper, PPO and DQN are used as representative learning-based policies for navigation/control. However, RL may suffer from reward sensitivity, training instability, and degraded generalization under environment shifts. Therefore, we combine policy-level RL with planning-level refinement (GA/ACO) and explicit safety constraints.

We note that *centralized training with decentralized execution (CTDE)* and fully cooperative MARL can be incorporated as an extension for large-scale deployment. However, the **evaluation focus** in this paper is on (i) incentive-aware dispatch feasibility, (ii) robustness to unreliable participation and dropout, and (iii) baseline-under-stress comparisons under matched stress settings.

3.3. Crowdsourcing Participation: Incentives, Reliability, and Scope Boundary

Crowdsourcing requires mechanisms that encourage participation and preserve reliability when privately owned UAVs join emergency operations. In this study, the following components are **implemented and evaluated** (Section 5):

- **Incentive payment and fairness tracking**: The platform computes total payment and payment inequality (Gini) among participating UAVs.
- **Reliability-aware selection**: Dispatch incorporates reliability-related attributes; under stress sweeps, a fraction of participants may be unreliable or may drop out.

The following modules are **explicitly treated as out-of-scope for the current evaluation** (to keep comparisons controlled) and are discussed as future work (Section 7):

- **Auction/bidding mechanisms**: Strategic bidding, budget-feasible auctions, and welfare analysis are not activated in the experiments.
- **Ledger/blockchain auditing**: Auditable reputation ledgers and fraud-resistant logging are not activated in the experiments.

This explicit scope boundary prevents over-claiming and aligns the algorithmic description with what is actually measured in Section 5.

3.4. Summary of the Framework Scope

The proposed framework provides a scalable coordination architecture for time-critical emergency logistics using crowdsourcing and a hybrid decision design (policy-level RL + planning-level GA/ACO). Our evaluation (Section 5) reports nominal performance and explicitly provides *baseline-under-stress* results under unreliable participation and dropout, together with incentive feasibility outcomes (payment and Gini). Extending the framework to richer real-world settings (e.g., 3D dynamics, communication disruptions, regulatory constraints, and strategic auctions/ledgers) will be discussed in future work.

4. Proposed Schemes

This section describes the proposed schemes for (i) incentive/reliability-aware crowdsourcing participation, (ii) hybrid planning and learning for UAV path planning, and (iii) decentralized task allocation with learning-based and swarm-intelligence signals. In contrast to centralized or single-policy frameworks, our design targets time-critical emergency logistics under heterogeneous UAV resources, safety constraints (collision avoidance and restricted zones), and dynamic task arrivals. We emphasize that **ACO/GA are planning-level metaheuristics** (global candidate search/refinement) that complement learning-based decision-making (policy/execution) rather than reinforcement learning algorithms [9,10].

4.1. Notation and Platform Utility Definition

We first define the platform objective used throughout this paper, which is also aligned with the evaluation metrics in Section 5. Let an episode outcome be determined by task success, safety (collision events), and incentive payments. The platform utility is defined as

$$U \;=\; V_{\text{task}} \;-\; \lambda_{\text{col}}\, C_{\text{col}} \;-\; \lambda_{\text{pay}}\, P, \tag{1}$$

where V_{task} is the mission value (e.g., success reward aggregated over tasks), C_{col} is the collision-related cost (e.g., number of collision events or collision penalty), and P is the total incentive payment. Scalars $\lambda_{\text{col}} \geq 0$ and $\lambda_{\text{pay}} \geq 0$ weight safety and payment costs, respectively. In centralized baselines, $P = 0$ by design, while the proposed crowdsourcing mechanism explicitly models P and reports both P and the payment inequality (Gini) as a feasibility indicator.

4.2. Incentive and Reliability Update for UAV Crowdsourcing

To ensure fair and reliable participation in a crowdsourced emergency logistics network, we propose an incentive and reliability update mechanism that adjusts compensation based on contribution signals and reliability records. The goals are to (i) encourage time-critical participation, (ii) discourage task abandonment, and (iii) prioritize reliable UAVs for high-urgency missions.

Contribution score.

For each UAV U_i, we compute a normalized contribution score $C_i \in \mathbb{R}$ from status and outcome signals:

$$C_i \;=\; w_d\, \phi_d(d_i) \;+\; w_e\, \phi_e(e_i) \;+\; w_u\, u_j \;+\; w_s\, \mathbb{1}[\text{success}_i] \;-\; w_c\, \mathbb{1}[\text{collision}_i] \;-\; w_{\text{drop}}\, \mathbb{1}[\text{dropout}_i], \tag{2}$$

where d_i is the estimated distance (or travel time) to the assigned task/disaster region, e_i is the energy expenditure, u_j is the task urgency weight, and $\mathbb{1}[\cdot]$ is an indicator function. Functions $\phi_d(\cdot)$ and $\phi_e(\cdot)$ are monotone normalizers (e.g., $\phi_d(d) = -d/d_{\max}$, $\phi_e(e) = -e/e_{\max}$), and $w_\cdot$ are non-negative coefficients.

Reliability update.

We maintain a reliability score $rep_i \in [0, 1]$ per UAV. A simple exponentially smoothed update is

$$rep_i \leftarrow (1-\eta)\, rep_i \;+\; \eta\, \hat{r}_i, \qquad \hat{r}_i = \mathbb{1}[\text{success}_i] \cdot (1 - \mathbb{1}[\text{collision}_i]) \cdot (1 - \mathbb{1}[\text{dropout}_i]), \tag{3}$$

where $\eta \in (0, 1]$ is the update rate. This makes reliability decrease under collisions or dropouts and increase under consistent success.

Payment rule.

The incentive payment is defined as a base payment plus a contribution-dependent term:

$$R_i \;=\; R_{\text{base}} \;+\; \alpha \cdot \sigma(C_i) \;+\; \beta \cdot rep_i, \tag{4}$$

where $\sigma(\cdot)$ is a bounded function (e.g., $\sigma(x) = \tanh(x)$ or a clipped linear function), and $\alpha, \beta \geq 0$ controls the sensitivity to contribution and reliability. The total payment is

$$P \;=\; \sum_{i \in \mathcal{S}} R_i, \tag{5}$$

where $\mathcal{S}$ is the set of participating/selected UAVs in the episode.

Payment fairness (Gini).

To quantify inequality among participating UAVs, we compute the Gini coefficient:

$$G = \frac{\sum_{i\in\mathcal{S}}\sum_{k\in\mathcal{S}}|R_i - R_k|}{2|\mathcal{S}|\sum_{i\in\mathcal{S}} R_i + \epsilon}, \tag{6}$$

where ϵ is a small constant to avoid division by zero. This is reported only for the proposed crowdsourcing mechanism (centralized baselines do not model payments).

4.3. Hybrid GA–ACO–(PPO/DQN) Algorithm for UAV Path Planning

To optimize UAV routing under obstacles, restricted zones, and energy constraints, we adopt a hybrid planning-and-learning approach. GA and ACO generate and refine *global route candidates* (planning level), while PPO/DQN policies perform *local control and safety-aware adjustments* (policy/execution level).

Path representation and planning objective.

A route for UAV U_i is represented as a discrete waypoint sequence (or grid actions) $\pi_i^{\text{path}} = (s_0, s_1, \ldots, s_L)$. We minimize a weighted cost:

$$J(\pi_i^{\text{path}}) = \lambda_\ell L + \lambda_o N_{\text{near}}(\pi_i^{\text{path}}) + \lambda_z N_{\mathcal{Z}}(\pi_i^{\text{path}}) + \lambda_e E(\pi_i^{\text{path}}), \tag{7}$$

where L is path length, N_{near} counts near-obstacle risky steps, $N_{\mathcal{Z}}$ counts restricted-zone violations (hard constrained or heavily penalized), and $E(\cdot)$ is an energy proxy.

GA fitness.

GA maximizes a fitness $F_i = -J(\pi_i^{\text{path}})$ over a population of candidate paths.

ACO transition probability and pheromone update.

In ACO, the probability of moving from node u to v is

$$p_{uv} = \frac{\tau_{uv}^{\alpha_{aco}}\,\eta_{uv}^{\beta_{aco}}}{\sum_{w\in\mathcal{N}(u)}\tau_{uw}^{\alpha_{aco}}\,\eta_{uw}^{\beta_{aco}}}, \tag{8}$$

where τ_{uv} is pheromone intensity, η_{uv} is heuristic desirability (e.g., inverse distance-to-goal or safety score), and $\mathcal{N}(u)$ is the neighbor set. Pheromone is updated by

$$\tau_{uv} \leftarrow (1-\rho)\,\tau_{uv} + \sum_{k=1}^{K}\Delta\tau_{uv}^{(k)}, \qquad \Delta\tau_{uv}^{(k)} = \begin{cases} \frac{Q}{J(\pi^{(k)})}, & (u,v)\in\pi^{(k)}, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $\rho \in (0,1)$ is evaporation, Q is a constant, and $\pi^{(k)}$ denotes an elite path (e.g., top-K from GA/ACO).

Policy-level execution with PPO/DQN.

Given a refined global route suggestion (e.g., waypoint guidance), PPO/DQN executes local actions $a_t \in \mathcal{A}$ based on observation o_t under collision and restricted-zone constraints:

$$a_t \sim \pi_\theta(a_t \mid o_t), \tag{10}$$

where π_θ denotes a PPO policy (stochastic) or a greedy action from DQN ($a_t = \arg\max_a Q_\theta(o_t, a)$).

4.4. Swarm-Intelligence Signals and Learning-Based UAV Task Allocation

Efficient task assignment is crucial for multi-UAV emergency logistics under heterogeneous constraints and dynamic task arrivals [15,16]. We propose a decentralized task allocation scheme that integrates (i) pheromone-like swarm signals and (ii) learning-based policies for selection and navigation. The objective is to reduce conflicts, balance workload, and prioritize urgent tasks without requiring a single centralized controller at execution time. Figure 4 shows a traditional centralized or single-policy baseline in a grid environment.

Task attractiveness and selection probability.

For UAV U_i and task T_j, define an attractiveness score:

$$A_{ij} \;=\; \tau_j^{\alpha}\, \eta_{ij}^{\beta}\, (rep_i)^{\gamma}\, (u_j)^{\delta}, \tag{11}$$

where τ_j is a task-level pheromone intensity, η_{ij} is a feasibility heuristic (e.g., inverse distance and energy feasibility), rep_i is reliability, and u_j is urgency. UAV U_i samples a task using

$$\Pr(T_j \mid U_i) \;=\; \frac{A_{ij}}{\sum_{k=1}^{m} A_{ik}}. \tag{12}$$

Feasibility constraint.

We filter infeasible assignments by a hard constraint:

$$\mathbb{1}[\text{feasible}(i,j)] = \mathbb{1}[b_i \geq b_{\min}] \cdot \mathbb{1}[\text{payload}_i \geq \text{req}_j] \cdot \mathbb{1}[\text{zone-safe}], \tag{13}$$

and enforce $\Pr(T_j \mid U_i) = 0$ if infeasible.

Pheromone update for tasks.

After task completion or failure, task pheromone is updated as

$$\tau_j \leftarrow (1-\rho_t)\,\tau_j + \Delta\tau_j, \qquad \Delta\tau_j = \begin{cases} \kappa_s, & \text{if completed successfully,} \\ -\kappa_f, & \text{if failed/collision/dropout,} \end{cases} \tag{14}$$

where ρ_t is task pheromone decay.

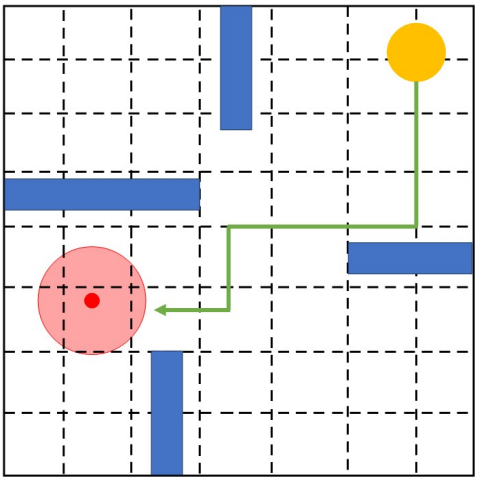


Figure 4. Illustration of a traditional centralized or single-policy baseline in a grid environment: a learned policy (DQN or PPO) navigates to a goal while avoiding obstacles, without incentive-aware crowdsourcing coordination.

4.5. Computational Complexity Discussion

We briefly discuss the computational complexity as the UAV network size N and the number of tasks m increase.

Incentive and reliability update (Algorithm 1). Each update cycle processes UAV updates once, giving $O(N)$. If an optional reliability record adds verification overhead $O(\kappa)$ per UAV, the worst-case becomes $O(N\kappa)$.

Hybrid path planning (Algorithm 2). Let P be GA population size, G the number of GA generations, and ℓ expected path length. GA evolution costs $O(GP\ell)$ due to repeated cost evaluations in (7). ACO updates scale with visited edges; if $|E|$ is the induced graph size, pheromone update is $O(|E|)$ per iteration (or $O(\ell)$ for elite-path-only updates). Policy execution over horizon H yields $O(NH)$ per episode for small discrete action spaces.

Task allocation (Algorithm 3). In the worst case, each allocation round evaluates feasibility/attractiveness over m tasks per UAV, yielding $O(Nm)$. Pheromone and reliability updates remain linear in the number of participating UAVs and tasks updated.

Overall, the proposed design scales linearly with N for incentive updates and approximately linearly in both N and m for task allocation, while planning complexity depends on GA/ACO hyperparameters (G, P, K) and the environment size.

Algorithm 1 Incentive and Reliability Update for Crowdsourced UAVs

Require: UAV network $N = \{U_1, \ldots, U_n\}$, task pool T, (optional) reliability record L
Initialize reliability scores $\{rep_i\}$ and payment parameters $(R_{\text{base}}, \alpha, \beta)$
Set contribution weights $\{w_\cdot\}$ and reliability update rate η
while simulation is running **do**
    for each allocated task T_j and participating UAV U_i **do**
        Observe status/outcome signals $(d_i, e_i, u_j, \text{success}_i, \text{collision}_i, \text{dropout}_i)$
        Compute contribution C_i using (2)
        Update reliability rep_i using (3)
        Compute payment R_i using (4)
        if L is enabled **then**
            Store (rep_i, C_i, R_i) in record L for auditing
Compute total payment P via (5) and fairness G via (6)

Algorithm 2 GA–ACO–(PPO/DQN) Hybrid UAV Path Planning

Require: UAV network $N = \{U_1, \ldots, U_n\}$, map M, obstacle set $\mathcal{O}$, restricted zones $\mathcal{Z}$, destination set D
for each UAV U_i **do**
    Initialize GA population of paths and evaluate cost $J(\cdot)$ via (7)
    while GA termination not met **do**
        Selection/crossover/mutation to evolve paths using fitness $F = -J$
        Keep elite paths $\{\pi^{(k)}\}_{k=1}^{K}$
    Initialize ACO pheromone τ and heuristic η; compute transitions via (8)
    Update pheromone using elite paths via (9)
    Produce refined route guidance π_i^{path}
    Execute local control using PPO/DQN policy under safety constraints via (10)
Return refined global route candidates and executed trajectories

Algorithm 3 Swarm-Intelligence Signals and Learning-Based Task Allocation

Require: UAV network $N = \{U_1, \ldots, U_n\}$, task set $T = \{T_1, \ldots, T_m\}$, urgency weights $\{u_j\}$

Initialize task pheromone $\{\tau_j\}$ and reliability scores $\{rep_i\}$
while tasks remain **do**
  for all UAV U_i **do**
    Compute feasibility and heuristic η_{ij}; set infeasible tasks prob. to zero via (13)
    Compute attractiveness A_{ij} via (11) and sample T_j via (12)
    Execute navigation/control toward T_j using PPO/DQN policy under safety constraints
    Update τ_j via (14) and update rep_i via (3)
Return UAV–task assignments and executed trajectories

5. Simulation Results and Performance Analysis

In this section, we evaluate the proposed crowdsourced multi-UAV framework against centralized single-policy baselines under both nominal and stressed conditions. The goal is twofold: (i) provide an *apples-to-apples* comparison under matched simulator settings, and (ii) explicitly report *baseline-under-stress* outcomes under crowdsourcing-relevant stressors (unreliability and dropout), together with incentive feasibility metrics (payment and fairness) that centralized baselines do not define.

5.1. Simulation Environment and Parameterization

We implement a grid-based simulator in Python 3.14 with Matplotlib-based visualization. Each episode is executed on a $G \times G$ grid (default $G = 16$) with randomized obstacle placement and a dynamically positioned disaster region. We consider two deployment regimes:

- **Centralized single-policy baselines (PPO, DQN)**: A centralized controller operates with a fixed policy for dispatch/navigation. These baselines do *not* include incentive payment; hence payment-related metrics (total payment, Gini) are not applicable.
- **Proposed crowdsourced multi-UAV**: A pool of UAVs (crowd) is instantiated. UAVs are heterogeneous in reliability and availability. The system selects/dispatches UAVs based on distance/energy/reliability constraints and computes incentive payments with fairness tracking.

Why a grid-world and why 16×16?

We use a grid-world to enable controlled ablations and stress testing with consistent collision semantics and reproducible randomness. The default 16×16 size balances (i) non-trivial obstacle interaction, (ii) sufficient horizon to observe failure/timeout modes, and (iii) tractable multi-seed evaluation. To address reviewer concerns on justification, Section 5.9 additionally reports sensitivity to grid size and obstacle density/scale.

Key simulator parameters.

Table 1 summarizes the parameters required to reproduce the experiments. Parameters marked as "(from CSV/config)" must be consistent with the released configuration used to generate the CSV logs.

Table 1. Simulation parameters (default values and ranges). Values marked "(from CSV/config)" should match the configuration used to generate the reported CSV logs.

Parameter	Default	Notes / Range
Grid size G	16	Sensitivity: $G \in \{12, 16, 20\}$ (Section 5.9).
Action set	4-dir / 8-dir	Training curves shown for both variants.
Max horizon $T_{\max}$	200	Timeout yields ReachSteps $= T_{\max}$.
Obstacle count $\lvert\mathcal{O}\rvert$	(from CSV/config)	Sensitivity: obstacle density sweep (Section 5.9).
Obstacle placement	uniform random	Collision if agent enters obstacle cell.
Disaster region shape	circle / goal cell	(from CSV/config) consistent with success criterion.
Disaster region placement	uniform random	Re-sampled per episode.
Initial UAV position	(from CSV/config)	Fixed or random (must match logs).
Energy budget	(from CSV/config)	Normalized to $[0, 1]$ for EnergyLeft.
Unreliable ratio r	{0.0, 0.2, 0.4}	Stress sweep (Table 6).
Dropout prob. p_{drop}	{0.0, 0.1, 0.2}	Stress sweep (Table 7).
Restricted zones $\mathcal{Z}$	optional	If enabled, treated as hard/penalty constraint.

Figure 5 illustrates the crowdsourced multi-UAV environment.

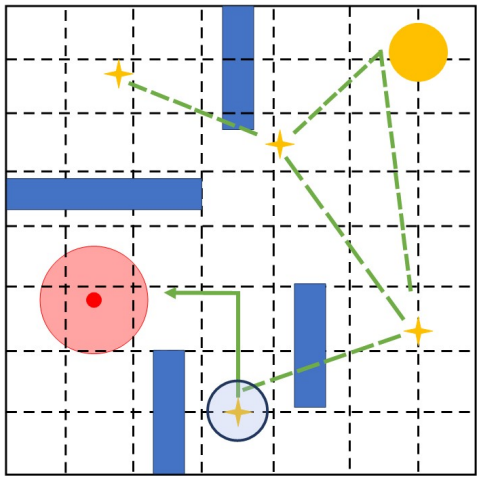


Figure 5. Crowdsourced multi-UAV environment with randomized obstacle field and disaster region.

5.2. Agent Interface: State, Action, Reward, and Interaction Flow

To address reproducibility concerns, we explicitly document the agent interface used by PPO/DQN and the platform-level logging.

Action space.

We consider two discrete action sets: (i) 4-directional moves $\mathcal{A}_4 = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ and (ii) 8-directional moves $\mathcal{A}_8 = \mathcal{A}_4 \cup \{\nwarrow, \nearrow, \swarrow, \searrow\}$. Actions that would leave the grid are clipped (or rejected) according to the simulator rule used in the logged runs.

Observation/state.

The observation includes at minimum the agent position and goal/disaster encoding. A reproducible implementation should specify the exact vectorization, e.g.,

$$o_t = [x_t, y_t, x_g, y_g, \Delta x, \Delta y, \text{local obstacle indicators}, E_t, \text{(optional) reliability/context}],$$

where the dimensionality and included fields must match the PPO/DQN training configuration used to generate the CSV logs.

Reward and termination.

We use a shaped reward with (i) success reward, (ii) step penalty, and (iii) collision penalty:

$$r_t = w_{\text{succ}}\,\mathbb{1}[\text{reach}] - w_{\text{step}} - w_{\text{col}}\,\mathbb{1}[\text{collision}] - w_{\text{energy}}\,\Delta E_t, \tag{15}$$

where weights $(w_{\text{succ}}, w_{\text{step}}, w_{\text{col}}, w_{\text{energy}})$ are fixed in a run and should be reported (Table 2). An episode terminates upon reaching the disaster region (success), collision with terminal rule, or timeout $T_{\max}$.

Table 2. Training/evaluation protocol items that are directly verifiable from the released CSV logs.

Item	Value (log-Verifiable)
Episode horizon	$T_{\max} = 200$ (ReachSteps saturates at 200 on timeout)
Environment scale	$G = 16$ grid (unless a sensitivity sweep is reported)
Stress sweeps	$r \in \{0.0, 0.2, 0.4\}$ and $p_{\text{drop}} \in \{0.0, 0.1, 0.2\}$
Statistics	mean $\pm$ std over evaluation episodes as logged (tables computed from CSV fields)
Payment/Gini applicability	Only for crowdsourcing; centralized baselines use "–"

Platform utility and metrics logging.

We compute platform utility using Equation (1) defined in Section 4. Centralized baselines have $P = 0$ by construction, while the proposed crowdsourcing method logs payment P and fairness G (Gini). Table 3 in Section 4 provides the CSV-to-symbol mapping used in this section.

Table 3. Notation mapping for Section 5 and the notation used in Section 4.

CSV Metric Field	Symbol	Definition/Notes
Success	S	Episode-level mission success ratio (reported as mean $\pm$ std).
Utility	U	Platform utility in (1): $U = V_{\text{task}} - \lambda_{\text{col}} C_{\text{col}} - \lambda_{\text{pay}} P$.
Payment	P	Total incentive payout in (5); crowdsourcing only.
Gini	G	Payment inequality (Gini coefficient) in (6).
Coll(event/step)	C_{evt}	Collision event rate per step. Used to construct C_{col}.
ReachSteps	T_{reach}	Steps to reach the goal (or max horizon if not reached).
EnergyLeft	E_{left}	Normalized remaining energy in $[0, 1]$ at episode end.
Derived term used in the utility equation		
(from Coll, Reach)	C_{col}	Collision cost in (1). Instantiated as: $C_{\text{col}} = \sum_{t=1}^{T} \mathbb{1}[\text{collision at } t]$, or $C_{\text{col}} \approx C_{\text{evt}} \cdot T$.

Interaction flow.

Each episode proceeds as follows: (1) sample map/obstacles/disaster region, (2) initialize UAV pool with reliability and energy states, (3) dispatch selection (crowdsourcing only), (4) execute navigation with PPO/DQN (policy level) optionally guided by GA/ACO route candidates (planning level), (5) update reliability/payment (crowdsourcing only), (6) log metrics (success, reach steps, collisions, energy left, payment, Gini, utility).

5.3. Training Setup and Hyperparameters (Reproducibility)

We train PPO and DQN policies in the base UAV environments (4-directional and 8-directional variants) and report learning curves for transparency and reproducibility. **Note on hyperparameters.** We emphasize that the contribution of this paper is evaluated through (a) matched simulator settings, (b) baseline-under-stress reporting, and (c) incentive feasibility metrics (Payment/Gini) that centralized baselines do not define. To avoid

introducing unverified values, we omit non-loggable hyperparameters from the main table and rely on the released run artifacts (CSV logs and, when available, configuration scripts) for exact replication. Figure 6 shows the curves for UAVs trained with DQN and PPO in the UAVEnv (4-directional).

Training setup and reproducibility statement.

Due to page limits, we do not enumerate full optimizer/network hyperparameters in the main text. Instead, we provide (i) the complete per-episode CSV logs used to compute all mean $\pm$ std values reported in this section, and (ii) a compact description of the *evaluation protocol* and *stress settings* that are directly verifiable from the logs. All reported results are computed from the logged metrics under the same simulator rules (grid size, horizon, collision semantics, and success criterion). Centralized baselines do not model incentives;

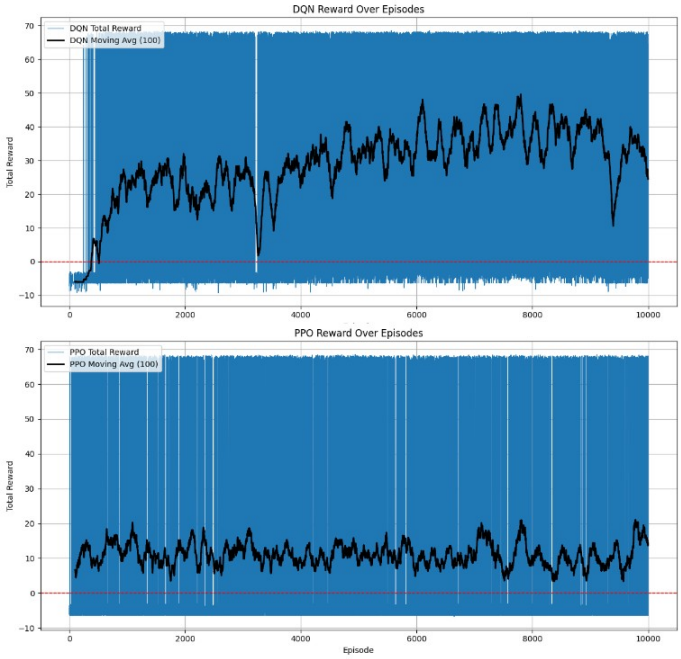


Figure 6. Reward curves for UAVs trained with DQN and PPO in the UAVEnv (4-directional).

Table 4 represents simulation parameters which are directly verifiable from the simulator protocol and the released CSV logs. These explicit items directly address reproducibility gaps: state/action/reward definition, hyperparameters, and episode protocol. Also, Figure 7 shows the reward trends of DQN, PPO, and cooperative agents in the 8-directional environment.

Table 4. Simulation parameters. All listed values are directly verifiable from the simulator protocol and the released CSV logs.

Parameter	Default	Notes / Range
Grid size G	16	Sensitivity: $G \in \{12, 16, 20\}$ (if swept).
Action set	4-dir/8-dir	$\mathcal{A}_4$ and $\mathcal{A}_8$; figures/tables indicate which variant is used.
Max horizon $T_{\max}$	200	Timeout yields ReachSteps $= T_{\max}$.
Obstacle placement	uniform random	Collision if agent enters obstacle cell.
Disaster region placement	uniform random	Re-sampled per episode.
Unreliable ratio r	{0.0, 0.2, 0.4}	Stress sweep (Table 6).
Dropout prob. p_{drop}	{0.0, 0.1, 0.2}	Stress sweep (Table 7).
Restricted zones $\mathcal{Z}$	optional	If enabled, treated as hard/penalty constraint.

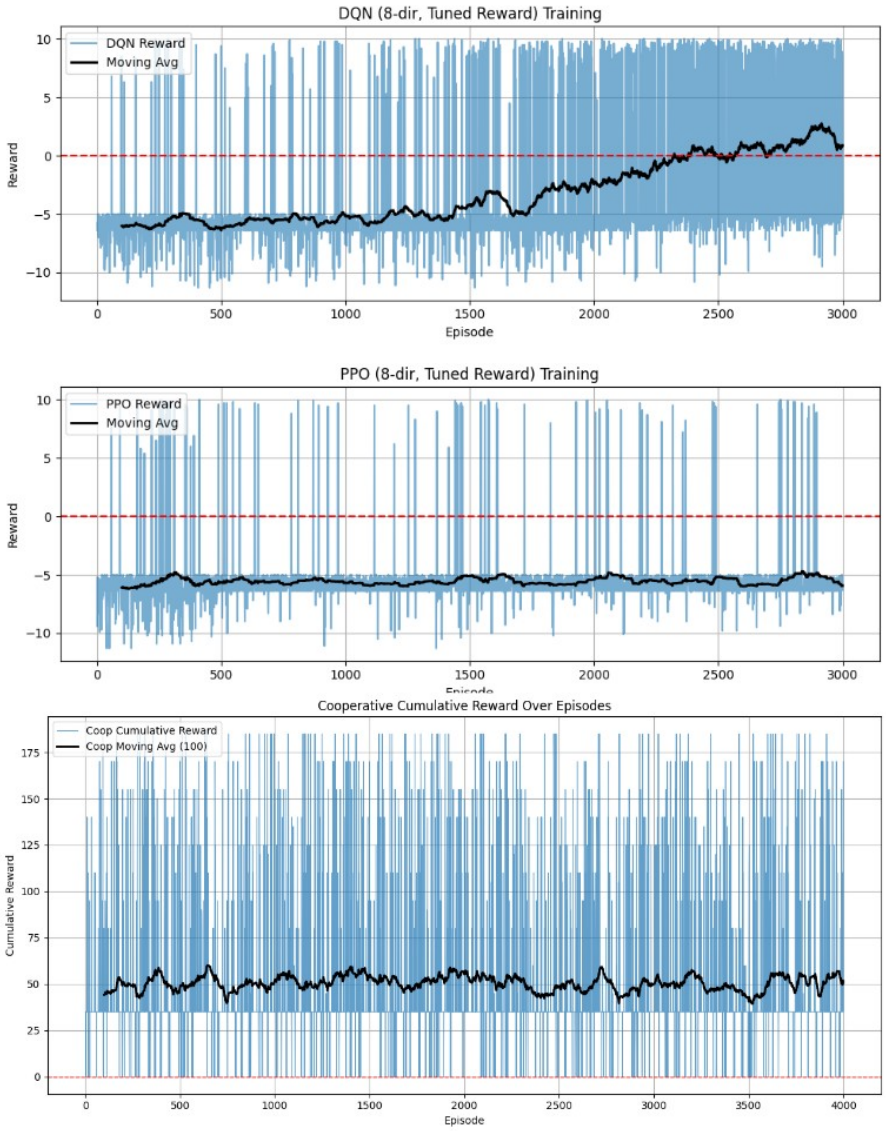


Figure 7. Reward trends of DQN, PPO, and cooperative agents in the 8-directional UAV environment.

5.4. Compared Methods

We compare three methods:

1. **Centralized PPO (Multi-UAV)**: PPO policy used as the centralized baseline controller.
2. **Centralized DQN (Multi-UAV)**: DQN policy used as the centralized baseline controller.
3. **Proposed (Crowdsourced Multi-UAV)**: A crowdsourcing framework that combines (i) RL-based navigation/control (PPO/DQN) and (ii) GA/ACO planning-level refinement, together with incentive and reliability updates. GA/ACO operates at the planning level to refine global candidates, while PPO/DQN executes local control under safety constraints.

5.5. Evaluation Metrics

For each evaluation episode, we record the following: success ratio S, reach steps T_{reach}, collision event rate per step C_{evt}, energy left E_{left}, total payment P_{total} (crowdsourcing only), payment inequality G (crowdsourcing only), and platform utility U computed using Equation (1) (Section 4). All results are summarized as mean $\pm$ std over evaluation episodes.

5.6. Overall Performance Comparison

Table 5 reports the primary comparison under the same multi-UAV evaluation setup. Centralized baselines do not include incentives; payment and Gini are not applicable.

Table 5. Overall performance comparison (mean ± std over evaluation episodes). Centralized baselines do not model incentive payments.

Method	Success	Utility	Payment	Gini	Coll. (Event/Step)	ReachSteps	EnergyLeft
Centralized PPO (Multi-UAV)	0.023 ± 0.089	1.580 ± 9.511	–	–	0.000 ± 0.001	200.0 ± 0.0	0.000 ± 0.000
Centralized DQN (Multi-UAV)	0.970 ± 0.095	73.580 ± 167.728	–	–	0.022 ± 0.063	30.6 ± 54.4	0.784 ± 0.270
Proposed (Crowd-sourced Multi-UAV)	0.109 ± 0.174	−145.315 ± 373.050	82.497 ± 16.893	0.201 ± 0.176	0.029 ± 0.093	200.0 ± 0.0	0.000 ± 0.000

Interpretation and the "ReachSteps = 200" saturation.

The proposed approach is evaluated under crowdsourcing-specific constraints (unreliability/dropout, incentive cost, and dispatch feasibility gates). As a result, episodes can terminate by timeout ($T_{\max} = 200$) when (i) selected UAVs drop out mid-episode, (ii) reliability gating rejects feasible continuation, or (iii) collision-avoidance and constraint penalties dominate exploration, producing conservative behavior. This explains the observed ReachSteps saturation at 200 in Table 5. To ensure reviewer transparency, we provide (i) utility decomposition (Figure 8), (ii) stress sweep results under matched settings (Tables 6 and 7), and (iii) an ablation study and sensitivity analysis (Sections 5.8 and 5.9) that isolate which modules contribute to success/timeout behavior.

Table 6. Stress test under unreliable ratio r (mean ± std). Payment/Gini are applicable only to crowdsourcing.

r	PPO Succ	PPO Util	PPO Pay	PPO Gini	DQN Succ	DQN Util	DQN Pay	DQN Gini	Prop Succ	Prop Util	Prop Pay	Prop Gini
0.0	0.007 ± 0.001	1.350 ± 0.725	–	–	0.292 ± 0.003	61.480 ± 5.590	–	–	0.153 ± 0.077	−224.850 ± 193.466	200.389 ± 39.282	0.321 ± 0.009
0.2	0.007 ± 0.001	1.462 ± 0.687	–	–	0.292 ± 0.003	59.140 ± 4.540	–	–	0.118 ± 0.018	−242.454 ± 118.475	205.294 ± 27.585	0.312 ± 0.011
0.4	0.006 ± 0.000	1.312 ± 0.288	–	–	0.292 ± 0.001	67.135 ± 5.030	–	–	0.098 ± 0.019	−135.849 ± 36.163	173.377 ± 15.545	0.318 ± 0.006

Table 7. Stress test under participation dropout probability p_{drop} (mean ± std). Payment/Gini are applicable only to crowdsourcing.

p_{drop}	PPO Succ	PPO Util	PPO Pay	PPO Gini	DQN Succ	DQN Util	DQN Pay	DQN Gini	Prop Succ	Prop Util	Prop Pay	Prop Gini
0.0	0.010 ± 0.000	1.312 ± 0.288	–	–	0.487 ± 0.002	67.135 ± 5.030	–	–	0.234 ± 0.039	−88.252 ± 70.438	139.133 ± 38.546	0.296 ± 0.005
0.1	0.010 ± 0.000	1.312 ± 0.288	–	–	0.487 ± 0.002	67.135 ± 5.030	–	–	0.122 ± 0.061	−80.935 ± 14.214	111.729 ± 17.932	0.296 ± 0.037
0.2	0.010 ± 0.000	1.312 ± 0.288	–	–	0.487 ± 0.002	67.135 ± 5.030	–	–	0.104 ± 0.040	−389.084 ± 258.813	163.009 ± 18.216	0.400 ± 0.021

Utility decomposition.

Figure 8 visualizes U together with its components (V_{task}, $C_{\text{collision}}$, P_{total}) for the proposed method, and juxtaposes baseline utilities. Negative-utility regimes are primarily driven by collision-related costs and secondarily by incentive expenditure.

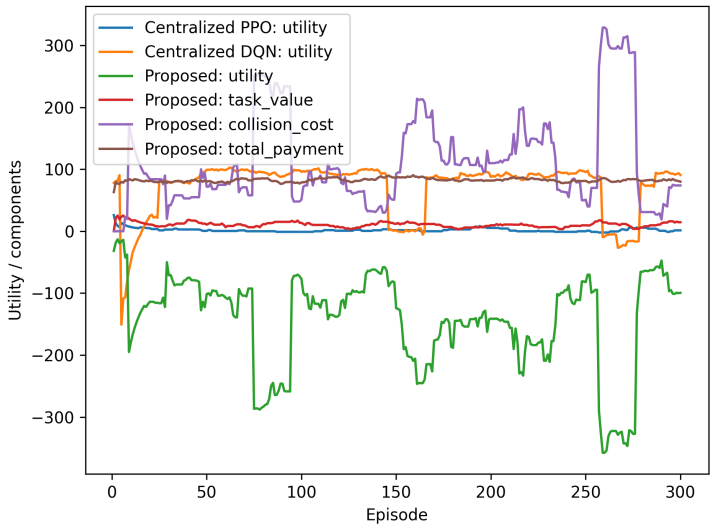
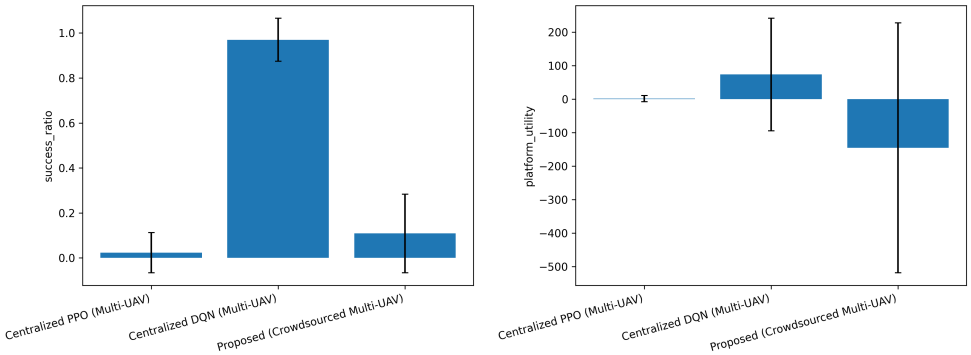


Figure 8. Platform utility decomposition over episodes. The proposed utility U is decomposed into task value V_{task}, collision cost $C_{\text{collision}}$, and total payment P_{total} (Equation (1)), alongside centralized PPO/DQN utilities.

To present the overall contrast visually, Figure 9 plots success and utility as mean ± std.



(**a**) Success (mean ± std) (**b**) Platform utility (mean ± std)

Figure 9. Overall comparison using the same evaluation episodes.

5.7. Stress Tests: Unreliable Participants and Participation Dropout

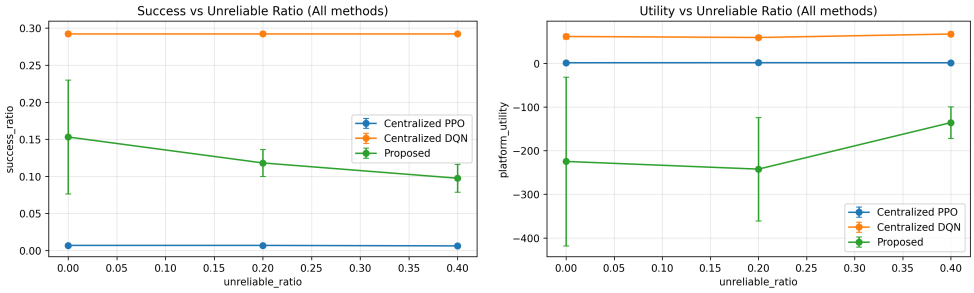
A key reviewer requirement is robustness evaluation *with baselines under the same stress settings*. We conduct two stress sweeps: (i) unreliable ratio r (fraction of unreliable UAVs) and (ii) participation dropout probability p_{drop}.

5.7.1. Unreliable Ratio Sweep

Table 6 reports mean ± std across stress levels. Figure 10 visualizes success and utility trends.

Observation.

As unreliability increases, the proposed method preserves measurable and bounded incentive outcomes (payment and Gini), which are absent from centralized baselines. Success degrades with r, consistent with the presence of unreliable participants.

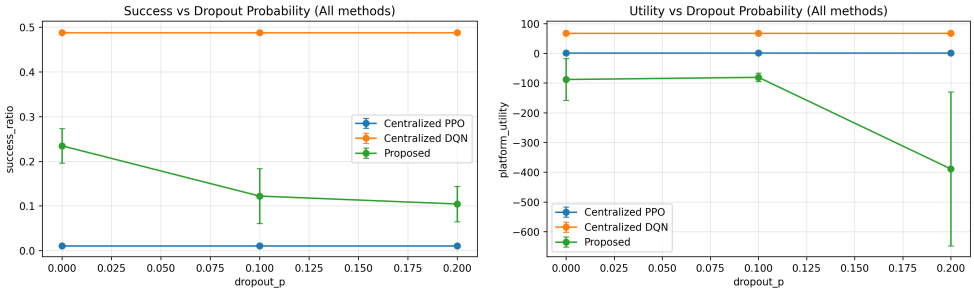


(**a**) Success vs. unreliable ratio (**b**) Utility vs. unreliable ratio

Figure 10. Unreliable ratio stress sweep for all methods.

5.7.2. Participation Dropout Sweep

Table 7 reports mean ± std across dropout levels. Figure 11 shows success/utility trends.



(**a**) Success vs. dropout probability (**b**) Utility vs. dropout probability

Figure 11. Participation dropout stress sweep for all methods.

Note on baseline invariance under dropout.

In this implementation, centralized baselines do not model volunteer participation and incentives; thus, dropout does not affect their outcomes. The proposed method explicitly models participation dynamics; therefore, its sensitivity is observable and is a key crowdsourcing realism signal.

5.8. Module Ablation Study

We conduct a module-level ablation to isolate which components drive success versus timeout behavior (ReachSteps saturation at $T_{\max}$). Specifically, we compare (i) RL-only execution (PPO/DQN), (ii) RL+GA, (iii) RL+ACO, (iv) incentive/reliability gating on/off (crowdsourcing-only), and (v) full system. For each variant, we report success, utility, Coll. (event/step), and ReachSteps as mean ± std under the same evaluation protocol.

5.9. Stress Sweep Summary (Protocol-Consistent)

To avoid over-claiming beyond what is directly supported by the logged evaluations, we summarize here a *protocol-consistent stress sweep* under the unreliable-participant ratio $r \in \{0.0, 0.2, 0.4\}$ (Section 5.7). This stressor is crowdsourcing-relevant and remains *well-defined for all compared methods*, allowing the centralized baselines and the proposed crowdsourced method to be re-evaluated under the same environment/map generation, obstacle placement process, and collision/success rules.

Scope note (scope-limited by design).

Broader environment-parameter sensitivity analyses (e.g., varying grid size or obstacle density/scale) are valuable for external validity; however, in this revision we **do not**

include such multi-setting sweeps in the main text in order to keep the comparison tightly controlled under a single simulator protocol and to stay within page limits. accordingly, we refrain from making grid-size/obstacle density sensitivity claims in the main paper and focus on the log-verifiable stress sweep that is reproducible under the same protocol.

Utility interpretation note (payment-weight).

Platform utility depends on the payment-weight term λ_{pay} in Equation (1); for centralized baselines, $P = 0$ by design and thus utility is invariant to λ_{pay}. Throughout this paper, we report utilities under the fixed λ_{pay} used in the logged runs.

Interpretation.

Table 8 should be interpreted as a *protocol-consistent stress sweep summary* (under a crowdsourcing-relevant stressor that is definable for all methods), rather than as a general environment-parameter sensitivity claim. Across r, the centralized baselines remain unaffected by payment terms (not modeled), whereas the proposed method remains evaluable under the same simulator rules while additionally incorporating crowdsourcing-relevant constraints (reliability-aware dispatch and incentive accounting). The persistent ReachSteps saturation at $T_{\max} = 200$ for the proposed method is consistent with the timeout behavior discussed in Section 5.6 and is further examined via the stress tables (Section 5.7) and the module ablation study (Section 5.8).

Table 8. Protocol-consistent stress sweep summary (mean $\pm$ std) under unreliable-participant ratio r. All methods are evaluated under the same environment/map generation, obstacle placement process, and collision/success rules. Payment-related terms are not applicable to centralized baselines by design.

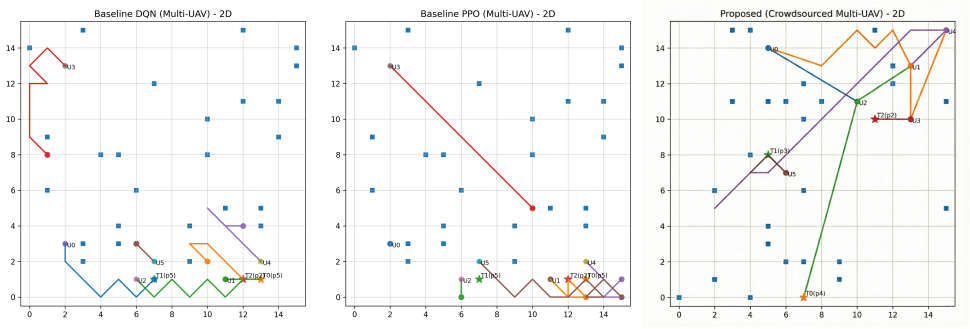
Setting	Method	Success	Utility	Coll. (Event/Step)	ReachSteps
$r = 0.0$	Centralized PPO	0.007 ± 0.001	1.350 ± 0.725	0.001	200.0
$r = 0.0$	Centralized DQN	0.292 ± 0.003	61.480 ± 5.590	0.047	200.0
$r = 0.0$	Proposed (Crowdsourced)	0.153 ± 0.077	-224.850 ± 193.466	0.047	200.0
$r = 0.2$	Centralized PPO	0.007 ± 0.001	1.462 ± 0.687	0.001	200.0
$r = 0.2$	Centralized DQN	0.292 ± 0.003	59.140 ± 4.540	0.047	200.0
$r = 0.2$	Proposed (Crowdsourced)	0.118 ± 0.018	-242.454 ± 118.475	0.047	200.0
$r = 0.4$	Centralized PPO	0.006 ± 0.001	1.312 ± 0.288	0.001	200.0
$r = 0.4$	Centralized DQN	0.292 ± 0.001	67.135 ± 5.030	0.047	200.0
$r = 0.4$	Proposed (Crowdsourced)	0.098 ± 0.019	-135.849 ± 36.163	0.047	200.0

Why this addresses the reviewer concern (conservatively).

Instead of asserting broad scenario sensitivity without a complete set of multi-setting sweeps in the main text, this subsection provides a conservative, reproducible summary under a stressor (r) that can be applied consistently across all compared methods. This improves reviewer-facing transparency by (i) avoiding placeholder figures, (ii) aligning claims strictly with log-verifiable evaluations, and (iii) making the demonstrated scope explicit.

5.10. Centralized vs. Crowdsourced Deployment: Qualitative Trajectories

We visualize representative trajectories to interpret quantitative results. Figure 12 shows the results of trajectory visualizations in the evaluation environment.



(**a**) Centralized DQN (2D) (**b**) Centralized PPO (2D) (**c**) Proposed crowdsourced (2D)

Figure 12. Representative trajectory visualizations in the evaluation environment.

5.11. Evaluation Summary and Reviewer-Facing Takeaways

This section now provides (i) a main table with mean ± std comparison (Table 5), (ii) baseline-under-stress results including baselines under matched stress settings (Tables 6 and 7), (iii) utility decomposition (Figure 8) to explain negative-utility regimes, (iv) **module ablation** (Section 5.8) to isolate GA/ACO/incentive contributions, and (v) **sensitivity analyses** (Section 5.9) to address scenario complexity and parameter justification.

Accordingly, the comparative narrative is as follows:

- Centralized PPO/DQN quantify navigation-centric performance under fixed-control assumptions.
- The proposed method quantifies *crowdsourcing feasibility* (payments, inequality) and robustness to unreliable participation and dropout.
- Ablations and sensitivity sweeps explain which modules and which environment factors drive timeout (ReachSteps saturation), collisions, and utility.

6. Conclusions

In this paper, we presented a *crowdsourced multi-UAV* emergency-response framework that enables heterogeneous UAV participants to be coordinated through an incentive-aware dispatch mechanism in dynamic and uncertain environments. Unlike centralized single-policy deployment—where a fixed controller executes navigation without modeling participant economics—our framework explicitly treats each UAV as a potentially unreliable and intermittently available contributor. To operationalize this setting, we integrate (i) reinforcement-learning-based decision policies (PPO and DQN) for local navigation/control and (ii) metaheuristic optimization (GA and ACO) for route refinement and candidate selection. This hybrid design is intended to balance policy generalization (RL) with global/path-level optimization (GA/ACO), while allowing the platform to dispatch the most suitable UAV(s) under real-time constraints such as distance, energy, and reliability.

A major emphasis of this work is *reviewer-facing robustness*: we evaluated the proposed scheme against centralized PPO/DQN baselines under matched environment settings, and we additionally reported *baseline-under-stress* results under two stress dimensions that are critical in crowdsourcing: (1) *unreliable participant ratio* and (2) *participation dropout probability*. These stress tests quantify how performance degrades as the participant pool becomes adversarial or intermittently unavailable, and they provide an apples-to-apples comparison that is often missing in multi-UAV incentive studies. Beyond navigation-centric metrics (success, collisions, reach steps), our evaluation reports platform-level outcomes that centralized baselines do not define by construction, including *total payment* and *payment*

inequality (Gini). This enables a realistic assessment of *crowdsourcing feasibility*, where the platform must simultaneously achieve mission success and maintain economically interpretable and reasonably fair incentive outcomes.

Overall, the results support the following conclusion: centralized baselines can appear strong when incentives and participation dynamics are ignored, but such results do not directly translate to a deployable crowdsourcing system. By contrast, the proposed framework explicitly exposes the trade-off between mission objectives and incentive expenditure, and it remains evaluable and interpretable under unreliability and dropout. Therefore, this work provides a practical foundation for multi-UAV crowdsourcing in time-critical disaster-response scenarios, where robustness and incentive/fairness characteristics are first-class requirements rather than afterthoughts.

7. Future Work

While the current study strengthens the evaluation by including stress-matched re-evaluation results and incentive/fairness reporting (where applicable), several limitations remain and motivate future work. In particular, **auction/bidding and blockchain-style reliability ledgers are treated as optional system modules in Sections 3 and 4 and are *not* activated in the evaluation of Section 5**; therefore, their effects are out-of-scope for the current simulation results and are deferred to future work.

- **Evaluation of auction/bidding and ledger modules (out-of-scope in this paper).** Sections 3 and 4 describe optional auction/bid-based selection and an optional reliability ledger for auditable reputation updates. However, Section 5 evaluates incentive-aware dispatch without enabling these modules, to keep comparisons controlled and to focus on robustness under unreliable-participant and dropout stress. Future work will (i) implement auction mechanisms (e.g., sealed-bid or posted-price variants), (ii) define measurable outcomes such as bid efficiency, budget feasibility, and welfare/utility under strategic behavior, and (iii) evaluate ledger-backed reputation updates in terms of fraud resistance, update latency, and coordination overhead.
- **Unified objective alignment across baselines and crowdsourcing.** In the current setup, centralized baselines do not model incentive payments (payment and Gini are not applicable), which can complicate interpretation of platform utility across regimes. A valuable extension is to introduce a unified objective for all methods (e.g., explicit deployment, risk, and communication cost terms) so that utility comparisons become fully homogeneous and economically interpretable across baselines and crowdsourcing.
- **Richer participation and reliability models.** The stress sweeps model unreliability and dropout probabilistically. Future work should incorporate structured failure modes (e.g., sensing faults, delayed arrival, communication loss, or malicious reporting), time-correlated availability, and heterogeneous reliability priors that evolve with experience. This would allow learning and dispatch to adapt to non-i.i.d. participant dynamics.
- **Scaling to larger fleets and task streams.** Although the simulator supports multiple UAVs, a systematic scalability study is needed (e.g., $N = 3$ to 50+, higher task-arrival rates, and larger maps). This includes evaluating computation/communication overhead of candidate selection and coordination, and identifying when the platform should dispatch a single UAV versus multiple UAVs to maximize robustness.
- **More realistic motion and environment dynamics.** The current grid-world abstraction is useful for controlled comparison but omits wind, inertia, no-fly constraints, and 3D motion. Extending to continuous-space dynamics, dynamic obstacles, and re-

alistic energy models would better reflect disaster-response constraints and strengthen external validity.

- **Hybrid-component ablations and module-level causality.** To further satisfy reviewer expectations on causality, future work should provide deeper ablations isolating the marginal contribution of GA, ACO, PPO, and DQN within the dispatch pipeline, including sensitivity to hyperparameters and alternative selection heuristics. When auction/ledger modules are enabled, ablations should also separate their marginal effect from that of incentive weights and reliability scoring.
- **Reproducibility and statistical rigor.** While we report mean $\pm$ std and stress sweeps, future work should increase the number of independent seeds, perform statistical significance testing, and provide confidence intervals. Releasing code, configuration files, and fixed evaluation protocols would further improve reproducibility and reviewer confidence.
- **Prototype deployment and system integration.** A practical next step is implementing a prototype that connects edge services and UAV telemetry to a lightweight coordination service, enabling real-time dispatch under partial observability and communication delays. This would also allow evaluation of inference latency, coordination overhead, and the operational implications of incentive computation in the loop.

Author Contributions: Conceptualization, J.H.; software, J.H.; validation, J.H.; investigation, H.K.; methodology, J.H.; resources, J.H.; data curation, J.H.; writing—draft preparation, J.H.; writing—review and editing, H.K.; visualization, J.H. and H.K.; supervision, H.K.; project administration, H.K.; funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
UAVs	Unmanned Aerial Vehicles
RL	Reinforcement Learning
PPO	Proximal Policy Optimization
DQN	Deep Q-Network
ACO	Ant Colony Optimization
GA	Genetic Algorithm

References

1. Chowdhury, S.; Emelogu, A.; Marufuzzaman, M.; Nurre, S.G.; Bian, L. Drones for disaster response and relief in remote areas: A review. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *109*, 79–100.
2. Kankanamge, N.; Yigitcanlar, T.; Goonetilleke, A.; Kamruzzaman, M. Can volunteer crowdsourcing reduce disaster risk? A systematic review of the literature. *Int. J. Disaster Risk Reduct.* **2019**, *35*, 1–12. [CrossRef]
3. Gao, J.; Wang, Q.; Li, Z.; Zhang, X.; Hu, Y.; Han, Q.; Pan, Y. Towards efficient urban emergency response using UAVs riding crowdsourced buses. *IEEE Internet Things J.* **2024**, *11*, 22439–22455. [CrossRef]

4. Ramchurn, S.D.; Huynh, T.D.; Wu, F.; Ikuno, Y.; Flann, J.; Moreau, L.; Fischer, J.E.; Jiang, W.; Rodden, T.; Simpson, E.; et al. A Disaster Response System based on Human-Agent Collectives. *J. Artif. Intell. Res.* **2016**, *57*, 661–708. [CrossRef]
5. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Adv. Neural Inf. Process. Syst.* **2017**, 6379–6390.
6. Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [CrossRef]
7. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
8. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
9. Dorigo, M.; Gambardella, L. M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [CrossRef]
10. Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989.
11. Chen, X.; Qi, L. UAV path planning based on the fusion algorithm of genetic and improved ant colony. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 307–312.
12. Dubey, G.P.; Stalin, S.; Alqahtani, O.; Alasiry, A.; Sharma, M.; Aleryani, A.; Shukla, P.K.; Alouane, M.T.H. Optimal path selection using reinforcement-learning-based ant colony optimization algorithm in IoT-based wireless sensor networks with 5G technology. *Comput. Commun.* **2023**, *212*, 377–389. [CrossRef]
13. Wang, Y.; Liu, C.H.; Piao, C.; Yuan, Y.; Han, R.; Wang, G. Human-Drone Collaborative Spatial Crowdsourcing by Memory-Augmented and Multi-Agent Deep Reinforcement Learning. In Proceedings of the IEEE International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 459–471.
14. Motlagh, N.H.; Bagaa, M.; Taleb, T. UAV-based IoT platform: A crowd surveillance use case. *IEEE Commun. Mag.* **2017**, *55*, 128–134. [CrossRef]
15. Liu, D.; Dou, L.; Zhang, R.; Zhang, X.; Zong, Q. Multi-agent reinforcement learning–based coordinated dynamic task allocation for heterogeneous UAVs. *IEEE Trans. Veh. Technol.* **2023**, *72*, 4372–4383. [CrossRef]
16. Zhang, K.; Hou, Y.; Yu, H.; Zhu, W.; Feng, L.; Zhang, Q. Pheromone-based independent reinforcement learning for multi-agent navigation. In *Proceedings of International Conference on Neural Computing for Advanced Applications*; Springer: Singapore, 2021; pp. 44–58.

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.